

02D8

#2

PATENT

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

Applicant(s): Hoon CHANG

Docket: 678-498

Serial No.: 09/580,683

Dated: July 6, 2000

Filed: May 30, 2000

For: APPARATUS AND METHOD FOR  
TRANSMITTING VARIABLE-LENGTH  
DATA ACCORDING TO A RADIO LINK  
PROTOCOL IN A MOBILE  
COMMUNICATION SYSTEM

Assistant Commissioner for Patents  
Washington, D.C. 20231

**TRANSMITTAL OF PRIORITY DOCUMENT**

Sir:

Enclosed is a certified copy of Korean Appln. No. 20081/1999 filed on  
May 27, 1999 and from which priority is claimed under 35 U.S.C. §119.

Respectfully submitted,

Paul J. Farrell  
Registration No. 33,494  
Attorney for Applicants

DILWORTH & BARRESE  
333 Earle Ovington Boulevard  
Uniondale, New York 11553  
(516) 228-8484

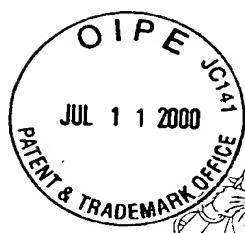
PJF/MM:mg

**CERTIFICATE OF MAILING UNDER 37 C.F.R. § 1.8(a)**

I hereby certify that this correspondence is being deposited with the United States Postal Service  
as first class mail, postpaid in an envelope, addressed to the: Commissioner of Patents and Trademarks,  
Washington, D.C. 20231 on July 6, 2000.

Dated: July 6, 2000  
Paul J. Farrell

8202-45



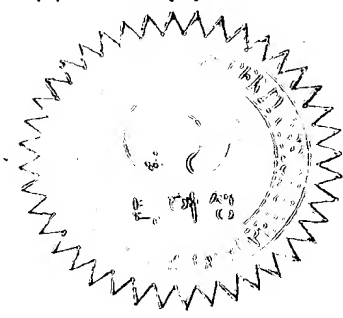
별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto is a true copy from the records of the Korean Industrial Property Office.

출원 번호 : 특허출원 1999년 제 20081 호  
Application Number

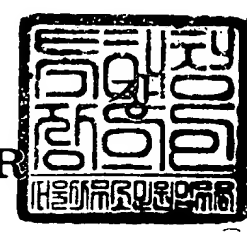
출원 년 월 일 : 1999년 05월 27일  
Date of Application

출원 인 : 삼성전자 주식회사  
Applicant(s)



2000 년 06 월 08 일

특 허 청  
COMMISSIONER





1019990020081

2000/6/

【서류명】	출원서
【권리구분】	특허
【수신처】	특허청장
【제출일자】	1999.05.27
【국제특허분류】	H04J
【국제특허분류】	H04L
【발명의 명칭】	이동통신시스템에서 라디오링크프로토콜에 따른 가변길이의 데이터 송수신 장치 및 방법
【발명의 영문명칭】	VARIABLE-LENGTH DATA TRANSMITTING AND RECEIVING APPARATUS AND METHOD IN ACCORDANCE WITH RADIO LINK PROTOCOL FOR MOBILE TELECOMMUNICATION SYSTEM
【출원인】	
【명칭】	삼성전자 주식회사
【출원인코드】	1-1998-104271-3
【대리인】	
【성명】	이건주
【대리인코드】	9-1998-000339-8
【포괄위임등록번호】	1999-006038-0
【발명자】	
【성명의 국문표기】	장훈
【성명의 영문표기】	CHANG, Hoon
【주민등록번호】	720929-1030128
【우편번호】	135-284
【주소】	서울특별시 강남구 대치4동 동아아파트 가동 1110호
【국적】	KR
【발명자】	
【성명의 국문표기】	이현석
【성명의 영문표기】	LEE, Hyun Seok
【주민등록번호】	690513-1227017
【우편번호】	463-020
【주소】	경기도 성남시 분당구 수내동 108-13
【국적】	KR

**【발명자】**

<b>【성명의 국문표기】</b>	김대균
<b>【성명의 영문표기】</b>	KIM,Dae Gyun
<b>【주민등록번호】</b>	681003-1690413
<b>【우편번호】</b>	135-240
<b>【주소】</b>	서울특별시 강남구 개포동 경남아파트 7동 905호
<b>【국적】</b>	KR

**【발명자】**

<b>【성명의 국문표기】</b>	구창회
<b>【성명의 영문표기】</b>	K00,Chang Hoi
<b>【주민등록번호】</b>	680620-1046313
<b>【우편번호】</b>	463-060
<b>【주소】</b>	경기도 성남시 분당구 이매동 한신아파트 205동 1105호
<b>【국적】</b>	KR

**【취지】** 특허법 제42조의 규정에 의하여 위와 같이 출원합니다. 대리인  
주 (인)

**【수수료】**

<b>【기본출원료】</b>	20 면	29,000 원
<b>【가산출원료】</b>	100 면	100,000 원
<b>【우선권주장료】</b>	0 건	0 원
<b>【심사청구료】</b>	0 항	0 원
<b>【합계】</b>	129,000 원	

**【첨부서류】** 1. 요약서·명세서(도면)\_1통



**【요약서】****【요약】**

이동 통신시스템에서 라디오링크프로토콜(RLP)에 따른 데이터를 송수신하는 장치 및 방법이 개시되어 있다. 이러한 본 발명의 데이터 송신장치는 송신용 데이터 버퍼, 다수의 서비스 제어기, 다중화/역다중화 제어기, 물리계층 제어기를 포함하여 이루어진다. 상기 다수의 서비스(RLP) 제어기 각각은 데이터 블록 요구에 대응하는 크기 내의 데이터를 데이터 버퍼로부터 읽어들이어 송신한다. 다중화/역다중화 제어기는 다수의 서비스 제어기중의 적어도 하나 이상의 서비스 제어기로부터의 데이터 블록을 포함하는 가변 길이의 정보 비트를 생성한다. 물리계층 처리기는 상기 생성된 정보 비트를 물리채널을 통해 송신한다.

**【대표도】**

도 7

**【색인어】**

Radio Link Protocol, Variable Length, Data Block, Information Bits, Sequence Number

**【명세서】****【발명의 명칭】**

이동 통신시스템에서 라디오링크프로토콜에 따른 가변 길이의 데이터 송수신 장치 및 방법 {VARIABLE-LENGTH DATA TRANSMITTING AND RECEIVING APPARATUS IN ACCORDANCE WITH RADIO LINK PROTOCOL FOR A MOBILE TELECOMMUNICATION SYSTEM AND METHOD THEREOF}

**【도면의 간단한 설명】**

도 1은 패킷데이터 서비스를 수행하는 일반적인 부호분할다중접속 통신시스템의 구성도.

도 2는 본 발명이 적용되는 RLP에 따른 데이터 송수신 장치의 구성도.

도 3은 본 발명에 따른 데이터 송신기의 구성도.

도 4는 본 발명에 따른 데이터 수신기의 구성도.

도 5는 본 발명에 따라 생성되는 프레임들의 포맷을 보여주는 도면.

도 6은 본 발명에 따라 생성되는 LTU의 포맷을 보여주는 도면.

도 7은 본 발명에 따라 생성되는 데이터 블록 및 RLP 프레임의 포맷을 보여주는 도면.

도 8은 본 발명에 따라 생성되는 RLP 프레임을 기본 채널을 통해 송수신할 시의 프레임 포맷을 보여주는 도면.

도 9는 본 발명에 따라 생성되는 RLP 프레임을 부가 채널을 통해 송수신할 시의 프레임 포맷을 보여주는 도면.



도 10은 본 발명에 따른 기본 채널 송신 동작의 처리 흐름도.

도 11은 본 발명에 따른 기본 채널 수신 동작의 처리 흐름도.

도 12는 본 발명에 따른 부가 채널 송신 동작의 처리 흐름도.

도 13은 본 발명에 따른 부가 채널 수신 동작의 처리 흐름도.

#### 【발명의 상세한 설명】

#### 【발명의 목적】

#### 【발명이 속하는 기술분야 및 그 분야의 종래기술】

<14> 본 발명은 부호분할다중접속(Code Division Multiple Access: 이하 CDMA라 칭한다) 방식의 이동 통신시스템에 관한 것으로, 특히 무선 환경에서 데이터의 효율적 전송을 위해 사용되는 라디오 링크 프로토콜(RLP: Radio Link Protocol)에 따라 데이터를 송수신하는 장치 및 방법에 관한 것이다.

<15> 일반적으로 CDMA방식의 이동 통신시스템은 음성을 위주로 하는 IS-95 규격에서 발전하여, 음성뿐만 아니라 고속 데이터의 전송이 가능한 CDMA2000 규격으로 발전하기에 이르렀다. 상기 CDMA2000 규격에서는 고품질의 음성, 동화상, 인터넷 검색 등의 서비스가 가능하다.

<16> 상기 CDMA2000 규격에서 제공하는 패킷 데이터 서비스의 예가 도 1에 나타나 있다. 상기 도 1에서 단말기는 터미널 장치(Terminal Equipment)인 TE와 단말 장치(Mobile Termination)인 MT로 이루어져 있다. 상기 도 1에서 기지국은 교환기

와 함께 BS/MSC(Base Station/Mobile Switching Center)로 표시되어 있고, 데이터 망(예: 인터넷)과 연동시켜주는 인터워킹펑션(Interworking Function)부는 IWF로 표시되어 있다. 상기 IWF부는 서로 다른 프로토콜을 사용하는 경우 상호간에 프로토콜 변환을 시켜 주는 기능을 수행하는 장치이다. 상기 도 1에서 단말기와 IWF부의 상위 서비스(Web Service)처리부들은 네트워크 프로토콜(IP: Internet Protocol)처리부와 링크 프로토콜(PPP: Point-to-Point Protocol)처리부를 거쳐서 데이터를 주고 받는다. 즉, 상위 서비스처리부들이 만든 데이터는 최종적으로 링크 프로토콜 패킷의 형태로 하위 계층에 전달되고, 하위 계층에서는 적절한 프로토콜을 사용하여 전송하게 된다.

<17>      상기 도 1에서는 TE와 MT 사이에 EIA-232제어기가 사용되는 예를 보여 주고 있다. EIA-232제어기를 통하여 MT에 전송된 링크 프로토콜 패킷들은 본 발명에서 다루는 라디오 링크 프로토콜(Radio Link Protocol: 이하 RLP라 칭한다)을 거쳐서 RLP 프레임(RLP Frame)에 나누어 들어간다. 상기 과정으로 생성된 RLP 프레임은 CDMA2000 규격인 IS-2000 규격을 통하여 연결된 물리 채널을 통하여 전송되게 된다. 연결된 물리 채널을 통하여 기지국으로 수신된 RLP 패킷은 다시 링크 프로토콜 패킷들로 복원이 되고 복원된 패킷들을 릴레이 계층을 통하여 상기 IWF부로 전송된다. 일반적으로 기지국과 IWF부 사이의 인터페이스는 IS-658 규격을 따른다. 상기 IWF부에서는 링크 프로토콜 패킷에서 데이터를 꺼내어 네트워크 프로토콜 처리부에게 전달하고 최종적으로 이러한 데이터들은 상위 서비스 처리부에게 전달된다.

<18>      상기 과정은 단말기에서 기지국으로 데이터를 전송하는 과정을 보여 주었지만 반대로 기지국에서 단말기에게 데이터를 전송하는 과정 또한 유사하게 진행된다.

<19>      CDMA2000에서는 다양한 서비스를 제공하기 위해서 상기 도 1과는 다른 여러 가지

구성이 가능하도록 되어 있다. 그러나 상위 서비스의 데이터를 담은 링크 프로토콜 패킷들은 모두 RLP를 통하여 무선 물리 채널로 전송된다는 점에서는 모두 공통된다.

<20> 현재 사용되는 RLP Type 3 규격은 현재 전송을 집합 1(Rate Set 1)에 대해서 9.6 kbps 또는 19.2 kbps의 물리 채널 프레임을 채울 수 있는 크기의 RLP 프레임, 또는 전송을 집합 2(Rate Set 2)에 대해서 14.4 kbps 또는 28.8 kbps의 물리 채널 프레임을 채울 수 있는 크기의 RLP 프레임만을 생성한다. 따라서 물리 채널이 153.6 kbps나 230.4 kbps와 같이 높은 전송율로 동작할 때는 상기 RLP 프레임 여러 개를 하나의 물리 채널 프레임에 넣는 방법을 사용한다. 만일 물리 채널이 상기 RLP Type 3 규격에서 지원하는 최대 전송율 153.6 또는 230.4 kbps 이상을 지원하는 경우, 예를 들어, 307.2 kbps, 460.8 kbps, 614.4 kbps, 1036.8 kbps 등을 지원하는 경우에는 더 많은 RLP 프레임을 하나의 물리 채널 프레임에 담게 된다. 그러나 이러한 방식은 하나의 물리 채널 프레임을 하나의 큰 크기의 RLP 프레임으로 채우는 방법에 비해서 프레임 헤더에 의한 부담이 커지고 사용하지 못하는 부분이 많아지기 때문에 효율적인 방법이 될 수 없다. 따라서 현재의 RLP Type 3 프레임 보다 더 큰 크기의 RLP 프레임을 전송하기 위해서는 기존의 방법과는 다른 새로운 방법이 필요하게 된다.

<21> 상기 RLP Type 3이 수행하는 중요한 일 가운데 하나는 조각 프레임을 생성하고 해석하는 일이다. 즉, 보내야 하는 재전송 프레임의 크기가 보낼 수 있는 데이터 블록의 크기 보다 크면 RLP는 상기 보내야하는 재전송 프레임의 데이터 부분을 여러 개의 조각 프레임(Segmented Frame)에 나누어서 전송하게 된다. RLP Type 3 규격에서는 상기 조각 과정(Segmentation Procedure)이 다음과 같이 이루어진다.

<22> 첫째, RLP Type 3은 조각을 최대 세 개까지 만들 수 있다. 각각의 조각은 FIRST,

SECOND, LAST의 세 가지 값으로 구별된다. 세 개 이상의 조각이 필요하면 RLP Type 3은 세 개의 조각 프레임으로 전송할 수 있을 때까지 상기 재전송 프레임을 전송하지 않는다

<23> 둘째, 조각 프레임에 대한 조립 작업(Reassembly Procedure)은 모든 조각이 수신되어야 가능하다. 즉, FIRST, SECOND, LAST 조각 프레임 중 어느 한 조각이라도 수신하지 못하면 전체 조각을 다시 전송하여야 한다.

<24> 상기 RLP Type 3 규격은 상기한 바와 같이 현재 전송율 집합 1(Rate Set 1)에 대해서 9.6 kbps 또는 19.2 kbps의 물리 채널 프레임을 채울 수 있는 크기의 RLP 프레임, 또는 전송율 집합 2(Rate Set 2)에 대해서 14.4 kbps 또는 28.8 kbps의 물리 채널 프레임을 채울 수 있는 크기의 RLP 프레임만을 생성하므로 전송율이 가장 낮을 때인 9.6 kbps 이나 14.4 kbps에서 최대 3 개의 조각으로도 충분히 전송이 가능하다. 그러나 물리 채널이 더 높은 전송율을 가지고 RLP에서 전송 효율을 높이기 위해 더 큰 크기의 RLP 프레임을 사용하게 되면 기존의 조각 방법과는 다른 새로운 방법이 필요하게 된다.

#### 【발명이 이루고자 하는 기술적 과제】

<25> 따라서 본 발명의 목적은 블록 단위 및 데이터 바이트 단위의 일련 번호를 사용함으로써 다양한 길이의 RLP 프레임을 가능하게 하여 다양한 크기와 구조를 갖는 물리 채널 프레임에 보다 많은 데이터를 전송할 수 있도록 하는데 있다.

<26> 본 발명의 또 다른 목적은 RLP 프레임을 전송할 때 블록 단위 및 데이터 바이트 단

위의 일련 번호를 사용함으로써 RLP 프레임 헤더의 비용을 줄여서 전송의 효율을 보다 높일 수 있도록 하는데 있다.

<27> 본 발명의 또 다른 목적은 상기 방법에서 발생하는 RLP 블록 또는 데이터 조각에 대한 재전송을 요청하는 NAK 프레임의 구성을 보다 효과적으로 할 수 있도록 하여 하나의 NAK 프레임으로 보다 많은 재전송 요청을 할 수 있도록 하는데 있다.

<28> 이러한 목적들을 달성하기 위한 본 발명은 상기와 같은 종래 기술의 문제점을 해결하기 위해서 RLP 프레임이 다양한 길이를 가질 수 있는 새로운 라디오 링크 프로토콜을 제안한다. 특히 본 발명에서는 데이터를 여러 개의 블록으로 묶고 블록마다 일련 번호를 부여함과 함께 블록에 속하는 데이터 바이트 각각에게도 일련 번호를 붙이는 방법으로 다양한 크기를 갖는 RLP 프레임에 가장 알맞은 크기의 헤더를 사용하여 전송하는 새로운 라디오 링크 프로토콜을 제안한다.

<29> 본 발명에 따른 데이터 송신장치는 송신용 데이터 버퍼, 다수의 서비스 제어기, 다중화/역다중화 제어기, 물리계층 제어기를 포함하여 이루어진다. 상기 다수의 서비스 (RLP) 제어기 각각은 데이터 블록 요청에 대응하는 크기 내의 데이터를 데이터 버퍼로부터 읽어들이어 송신한다. 다중화/역다중화 제어기는 다수의 서비스 제어기중의 적어도 하나 이상의 서비스 제어기로부터의 데이터 블록을 포함하는 가변 길이의 정보 비트를 생성한다. 물리계층 처리기는 상기 생성된 정보 비트를 물리채널을 통해 송신한다.

#### 【발명의 구성 및 작용】

<30> 이하 본 발명의 바람직한 실시예의 상세한 설명이 첨부된 도면들을 참조하여 설명

될 것이다. 우선 각 도면의 구성요소들에 참조부호를 부가함에 있어서, 동일한 구성요소들에 대해서는 비록 다른 도면상에 표시되더라도 가능한 동일한 부호를 가지도록 하고 있음에 유의해야 한다.

- <31>        도 2는 본 발명이 적용되는 RLP에 따라 데이터를 송수신하는 이동 통신시스템의 구성을 보여주는 도면이다.
- <32>        상기 도 2에서 물리계층 처리기 150,250은 IS-2000 규격에 따라서 단말기와 기지국 사이에 물리 채널을 연결하고, RLP처리기 130,230에서 전달되는 RLP 프레임들을 상기 연결된 물리 채널을 통하여 상대방 물리 계층에게 전달하고, 상기 물리 채널을 통하여 수신된 RLP 프레임들을 RLP처리기 130,230에게 전달하는 구성 요소들을 하나의 블록으로 나타낸 것이다.
- <33>        다중화/역다중화 제어기 140,240은 물리계층 처리기 150,250에게 전달할 RLP 프레임들에 대해서 목적지와 크기 등의 정보를 RLP 프레임들 앞에 붙인 후 물리계층 처리기 150,250에게 전달하는 다중화 기능과, 반대로 수신된 RLP 프레임에 대해서 목적지와 크기를 확인한 후 상위 RLP처리기 130,230에게 전달하는 역다중화 기능을 수행한다.
- <34>        송신 및 수신 데이터 버퍼 122,124,222,224는 메모리 장치로서 도 1의 예에서와 같이 EIA-232 제어기나 IS-658 제어기가 상기 링크 프로토콜 처리기 110,210에서 받은 데이터를 저장해둔다. 상기 데이터 버퍼들 122,222는 RLP처리기 130,230가 요구할 때마다 저장된 패킷을 순서대로 요구된 크기에 맞게 잘라준다. 또한 상기 데이터 버퍼들 124,224는 RLP처리기 130,230에서 데이터를 전달할 때마다 순서대로 저장해둔다. 이 저장된 데이터는 상기 EIA-232 제어기나 IS-658 제어기에 의해 PPP처리기 또는 IWF부에게 전달된다. 상기 EIA-232 제어기나 IS-658 제어기는 EIA-232 규격과 IS-658 규격에 따라



동작하며, 상기 데이터 버퍼들 122,124,222,224와 링크 프로토콜 처리기 110,210 사이에서 데이터 송수신을 맡는다. 현재 CDMA2000 패킷 서비스에서는 상기 EIA-232 제어기나 IS-658 제어기 이외에 다른 방식을 사용할 수도 있으므로, 상기 도 2에서는 이들 제어를 표시하지는 않았다는 사실에 유의할 필요가 있다.

<35> 도 3은 본 발명에 따른 데이터 송신기의 구성을 보여주는 도면이다.

<36> 도 3을 참조하면, RLP 프레임을 송신 처리하기 위한 RLP 처리기 130은 RLP 제어기 131과, L\_V(S) 레지스터 132와, 재전송 버퍼 133을 포함한다. 상기 RLP 제어기 131은 송신 데이터 버퍼 122에서 데이터를 받아서 RLP 프레임을 생성한 다음, 다중화/역다중화 제어기 140에게 이 생성된 RLP 프레임을 담은 데이터 블록을 전송한다. 상기 재전송 버퍼(Forward Resequencing Buffer) 133은 메모리 장치로서 재전송을 위한 데이터를 기억시키기 위해서 사용된다. 상기 L\_V(S) 레지스터 132는 상기 RLP 제어기 131이 송신 데이터 버퍼 122에 저장되어 있는 송신을 위한 데이터를 여러개의 블록으로 묶고, 블록마다 일련번호('블록 일련번호')를 부여하고 또한 각 블록에 데이터 바이트 각각에게 일련번호를 붙일 시 사용될 일련번호('데이터 일련번호')를 카운트하는 역할을 한다.

<37> 도 4는 본 발명에 따른 데이터 수신기의 구성을 보여주는 도면이다.

<38> 도 4를 참조하면, RLP 프레임을 수신 처리하기 위한 RLP 처리기 130은 RLP 제어기 131과, L\_V(N), L\_V(R) 두 레지스터 135,136과, 레지스터 E 134와, NAK(Non Acknowledge) 리스트 137, 재정렬 버퍼 138을 포함한다. 상기 RLP 제어기 131은 다중화/역다중화 제어기 140에서 RLP 프레임을 받아서 순서에 맞는 데이터가 도착하였는지 살펴본다. 이때 만일 순서에 맞다면 수신 데이터 버퍼 124에 데이터를 전달한다. 만일 그렇지 않다면 재정렬 버퍼 138에 저장한 다음, NAK 리스트 137에 재전송을 요청할 부분을 기록하고 다음번

제어 프레임을 송신할 때 NAK 리스트 137에 저장되었던 부분을 넣어서 보낸다.

<39>      상기 레지스터 E 134는 깨진 데이터 블록의 수를 기록하기 위한 것이다. 상기 다중화/역다중화 제어기 140에서 깨진 데이터 블록을 알리면, RLP 제어기 131은 이 값을 레지스터 E 134에 기록한 후, 재설정이 필요할 때 사용할 수 있다. 상기 레지스터 L\_V(N) 135는 수신되어야 할 데이터중 수신하지 못한 첫 번째 데이터 바이트의 일련번호를 저장한다. 상기 레지스터 L\_V(R) 136은 다음에 새로이 수신되어야 할 데이터 바이트의 일련번호를 저장한다. 예를 들어, 1번 바이트에서부터 7번 바이트까지의 데이터가 수신되었고 이때 3번 및 4번 바이트의 데이터가 깨져서 수신되지 못하였다면, 상기 레지스터 L\_V(N) 135에는 수신하지 못한 첫 번째 데이터 바이트의 일련번호인 3번이 저장되고, 상기 레지스터 L\_V(R) 136에는 새로이 수신되어야 할 데이터 바이트의 일련번호인 8번이 저장되게 된다.

<40>      본 발명에 따라 가변길이의 RLP 프레임을 생성하고 송수신하는 동작은 크게 다중화/역다중화 제어기 140,240과, RLP 처리기 130,230의 동작으로 대별되어 되어 설명될 수 있을 것이다. 상기 다중화/역다중화 제어기 140과 240의 동작은 동일하게 수행되고, RLP 처리기 130과 230의 동작은 동일하게 수행되므로, 하기에서의 본 발명에 따른 동작은 편의상 다중화/역다중화 제어기 140과, RLP 처리기 130의 동작으로 국한되어 설명될 것이다.

<41>      다중화/역다중화 제어기의 송신동작

<42> 현재 연결되어 있는 물리 채널을 통하여 패킷 데이터 뿐만 아니라 음성 데이터나, 기타 다른 여러 가지 정보들이 동시에 송신될 수 있으므로, 하기에서는 다중화/역다중화 제어기에게 전송할 데이터를 제공하는 역할을 수행하는 단위를 '서비스' 또는 '서비스 블록'이라고 칭할 것이다. 또한 다중화/역다중화 제어기 140과 물리계층 처리기 150이 서로 주고 받는 전송 단위를 '정보 비트(information bits)' 또는 '물리프레임(physical frame)의 페이로드(payload)'라고 칭하고, RLP처리기 130을 포함한 상위 서비스 블록과 다중화/역다중화 제어기 140이 서로 주고 받는 전송 단위를 '데이터 블록(data block) 또는 RLP프레임'이라고 칭할 것이다.

<43> 송신측 다중화/역다중화 제어기 140은 적어도 매 설정시간(예: 20 ms)마다 물리계층 처리기 150에게 전송할 정보 비트를 생성하여 전달하여야 한다. 즉, 다중화/역다중화 제어기 140은 현재 연결되어 있는 모든 물리 채널에 대해서 물리 채널로 전송되는 프레임의 페이로드(payload)를 채울 정보 비트를 생성하여 전달하여야 한다. IS-2000 규격에는 기본 채널(Fundamental Channel), 전용 제어 채널(Dedicated Control Channel), 부가 채널(Supplemental Channel) 등이 존재한다. 상기 다중화/역다중화 제어기 140이 이러한 정보 비트를 생성하여 물리 계층 처리기 150에게 전달하여 상기 기본 채널, 전용 제어 채널, 또는 부가 채널 중 어느 한 채널로 전송하고자 할 때는 다음과 같은 값들을 전달한다.

<44> - SDU: 실제 전송될 정보 비트를 담는다. 전용 제어 채널 또는 부가 채널의 경우 만일 전송할 정보 비트가 없다면 상기 다중화/역다중화 제어기와 물리 계층이 미리 약속한 널값(NULL)을 채운다.

<45> - FRAME\_SIZE: 정보 비트가 담길 물리 채널 프레임의 크기를 담는다. 상기 SDU에

널값이 지정되는 경우에는 물리 계층에서 이 값을 무시한다.

<46> - FRAME\_RATE: 정보 비트가 담길 물리 채널 프레임이 전송될 전송 속도를

지정한다. 상기 SDU에 널값이 지정되는 경우에는 물리 계층에서 이 값을 무시한다.

<47> 송신측 다중화/역다중화 제어기 140이 상기한 내용을 물리계층 처리기 150에게 전달하면, 상기 물리계층 처리기 150은 지정된 부호화 방법 및 변조 방식 등을 사용하여 전달된 내용을 처리한 후 수신측에게 전송한다.

<48> 송신측 다중화/역다중화 제어기 140은 물리 채널로 전달할 논리 전송 단위의 페이로드 또는 정보 비트를 생성하기 위해, 논리적 채널(logical channel)이 현재 연결되어 있는 물리 채널에 대응되는 서비스들에서 전송할 데이터 블록(data block)을 사용한다. 상기 논리적 채널이 연결된 물리 채널에 대응되는 서비스는 현재 생성되는 정보 비트를 전송할 물리 채널로 자신의 데이터 블록을 전송할 수 있는 서비스를 말한다. 이러한 서비스를 단말과 기지국 상호간에 서로 연결하고, 또한 상기 서비스의 논리적 채널을 물리 채널 위에 연결시키는 과정은 IS-2000에 규정된 시그널링 메시지(signaling message)와 시그널링 절차(signaling procedure)로 가능하다.

<49> 송신측 다중화/역다중화 제어기 140은 현재 논리적 채널이 연결되어 있는 물리 채널에 대응되는 서비스들의 데이터 블록을 전송하기로 결정하면, 적절한 우선 순위에 따라 상기 서비스에게서 적절한 길이의 데이터 블록(도 5의 (a) 참조)을 받아들인다. 상기 다중화/역다중화 제어기 140은 상기 서비스에게서 데이터 블록을 받아들일 때 수신측 다중화/역다중화 제어기에서 수신된 상기 데이터 블록을 전달할 서비스를 알 수 있도록 서비스 인식자(service identifier)와, 길이 정보를 상기 데이터 블록에 덧붙인 MuxPDU를 만든다(도 5의 (b) 참조). 상기 MuxPDU는 여러 서비스에서 받은 여러 개의 데이터 블

록과 시그널링 메시지를 함께 가질 수도 있다. 상기 정보 비트는 이러한 MuxPDU 하나 또는 여러 개를 포함하고 있으며, 또한 하나 또는 여러 개의 MuxPDU 마다 오류를 발견할 수 있는 CRC(Cyclic Redundancy Code)가 추가될 수 있다. 여러 개의 MuxPDU 마다 오류를 발견할 수 있는 CRC가 추가되는 경우, 하나의 CRC와 상기 CRC로 보호되는 정보 비트의 일부분을 본 발명에서는 '논리 전송 단위(LTU: Logical Transmission Unit)'이라고 칭하고 상기 물리 계층에게 전달될 정보 비트를 여러 부분으로 나누어서 각 나누어진 부분마다 오류를 검사할 수 있는 CRC를 넣는 방식을 '논리 전송 단위(logical transmission unit)가 사용되었다'고 한다. 이때 상기 CRC에 의해서 보호되는 CRC를 제외한 논리 전송 단위의 나머지 부분을 '논리 전송 단위의 페이로드'라고 칭할 것이다(도 5의 (c)참조). 이들 논리 전송 단위는 수신측 다중화/역다중화 제어기에서 올바르게 수신되었는지의 여부를 판단하는 기본 단위가 된다. 만일 상기 논리 전송 단위가 사용되지 않는다면 수신측 다중화/역다중화 제어기에서 올바르게 수신되었는지 판단하는 기본 단위는 정보 비트가 된다.

<50> 송신측 다중화/역다중화 제어기 140은 현재 전송될 물리 채널 프레임에 대해서 가능한 전송율과 정보 비트의 크기, 그리고 논리 전송 단위가 사용되는지 여부와 사용될 경우의 논리 전송 단위 크기와 CRC 생성 방법 등을 미리 알고 있어야 한다. 이러한 '형상 정보(configuration)'는 단말과 기지국이 미리 약속한 범위에서, 물리 계층 처리기 150이 제공하는 현재의 물리 채널 상태에 따라 상기 송신측 다중화/역다중화 제어기 140이 생성할 정보 비트의 크기와 논리 전송 단위의 생성 방식을 결정하는데 사용한다.

<51> 상기 송신측 다중화/역다중화 제어기 140은 논리 전송 단위의 사용을 결정하면 데이터 블록을 포함하는 MuxPDU로 논리 전송 단위의 페이로드를 채우고 남은 공

간은 채움 MuxPDU 또는 채움 비트열로 채운 다음, 생성된 논리 전송 단위의 페이로드에 대한 CRC를 생성한다. 상기 송신측 다중화/역다중화 제어기 140은 상기 과정을 필요한 논리 전송 단위 수만큼 반복하고 생성된 논리 전송 단위를 차례차례 집어넣어 정보 비트를 채운 다음, 남은 공간은 '0'으로 채우고 물리 채널 제어기 150에게 전달한다.

<52>       상기 송신측 다중화/역다중화 제어기 140은 논리 전송 단위를 사용하지 않기로 결정하면 데이터 블록을 포함하는 MuxPDU로 채우고 정보 비트를 남은 공간은 채움 MuxPDU 하나 또는 채움 비트열로 채운 다음, 생성된 정보 비트를 물리 채널에게 전달한다.

<53>       상기 송신측 다중화/역다중화 제어기 140은 더 이상의 전송할 데이터 블록이 없는 경우, 정보 비트의 남은 부분을 채우기 위해서 수신측 다중화/역다중화 제어기와 미리 약속한 특정 서비스 인식자로 설정한 MuxPDU를 사용하거나 또는 수신측 다중화/역다중화 제어기와 미리 약속한 일정한 비트열(bit pattern)을 사용하여 정보 비트의 남은 부분을 채운다. 본 발명에서는 상기 특정 서비스 인식자로 설정한 MuxPDU를 '채움 MuxPDU(fill MuxPDU)', 그리고 상기 일정한 비트열을 '채움 비트열(fill bit pattern)'이라고 칭할 것이다.

<54>       송신측 다중화/역다중화 제어기 140은 상기 과정에서 논리적 채널이 연결된 물리 채널에 대응되는 모든 서비스와 시그널링 메시지 생성기에서 전송할 아무런 시그널링 메시지가 데이터 블록을 받지 못했을 때는 현재 전송하고자 하는 물리 채널에 따라서 다르게 동작한다. 즉, 전용 제어 채널이나 부가 채널의 경우 널값(NULL)을 SDU로 전달한다. 기본 채널의 경우에는 수신측 다중화/역다중화 제어기와 미리 약속한 일정한 비트열을 정보 비트로 하여 물리 채널에 전달한다. 본 발명에서는 상기 일정한 비트열을 '널 트래픽(null traffic)'이라고 칭할 것이다.

<55>      상기 서비스 블록은 송신측 다중화/역다중화 제어기에게 서비스가 아무런 데이터 블록도 전송할 것이 없음을 알리는 방법으로 '빈 데이터 블록(blank data block)'을 전달한다. 상기 빈 데이터 블록은 아무런 내용도 가지지 않는 데이터 블록으로 상기 특수한 목적을 위해서만 사용된다.

<56>      다중화/역다중화 제어기의 수신동작

<57>      수신측 물리계층 처리기 150은 수신되는 신호를 지정된 복호화 방법 및 복조 방식 등을 사용하여 해석한 다음, 수신된 물리 계층 프레임에 담긴 정보 비트 부분을 수신측 다중화/역다중화 제어기 140에게 전달한다. 물리계층 처리기 150이 이러한 정보 비트를 해석하여 다중화/역다중화 제어기 140에게 전달할 때는 다음과 같은 정보가 포함되어 전달된다.

<58>      - SDU: 실제 수신된 정보 비트를 담는다. 만일 수신된 정보 비트가 없거나 프레임이 깨졌으면, 상기 다중화/역다중화 제어기 140과, 물리계층 처리기 150이 미리 약속한 널값(NULL)을 채운다.

<59>      - FRAME\_QUALITY: 수신된 물리 채널 프레임이 올바른(valid) 프레임인지 아닌지의 여부를 알려준다.

<60>      - FRAME\_SIZE: 수신된 물리 채널 프레임의 크기를 담는다. 이 값은 수신된 물리 채널 프레임의 전송 속도에 따라서 결정된다.

<61>      - FRAME\_RATE: 수신된 물리 채널 프레임의 전송 속도를 담는다.

- <62>      상기 수신측 다중화/역다중화 제어기 140은 현재 수신된 물리 채널 프레임에 대해서 전송율과 정보 비트의 크기, 그리고 논리 전송 단위가 사용되었는지 여부와, 논리 전송 단위가 사용되었다면 수신된 논리 전송 단위의 크기, CRC 생성 방법 등을 미리 알고 있어야 한다. 이러한 형상 정보(configuration)은 단말과 기지국이 미리 약속한 범위에서 물리계층 처리기 150이 제공하는 상기 정보에 따라 결정할 수 있는 것으로 한다.
- <63>      상기 수신측 다중화/역다중화 제어기 140은 만일 수신측 물리계층 처리기 150에서 아무런 물리 채널 프레임도 수신되지 않았다고 판단되어 SDU에 널값을 채우고 FRAME\_QUALITY에 올바른 프레임이 수신되었다고 알려주면, 논리적 채널이 연결된 물리 채널에 대응되는 모든 서비스들에게 아무 프레임도 수신되지 않았다고 알린다.
- <64>      상기 수신측 다중화/역다중화 제어기 140은 만일 수신측 물리계층 처리기 150에서 SDU에 널값을 채우지 않았거나 FRAME\_QUALITY에 깨진 프레임이 수신되었다고 알려주면 상기 형상 정보와 수신측 물리계층 처리기 150이 제공한 정보에 의해서 논리 전송 단위가 수신된 프레임에 사용되었는지 결정한다.
- <65>      상기 수신측 다중화/역다중화 제어기 140은 만일 논리 전송 단위가 사용되었다면 논리 전송 단위의 길이와 CRC 검사 방법, 그리고 논리 전송 단위의 개수를 결정한다. 상기 다중화/역다중화 제어기 140은 논리 전송 단위의 개수 만큼 수신된 정보 비트에서 논리 전송 단위를 분리한다. 상기 다중화/역다중화 제어기 140은 이미 논리 전송 단위의 길이와 개수를 알고 있으므로 수신된 정보 비트에서 상기 길이 단위로 영역을 상기 개수 만큼 나누어서 논리 전송 단위를 구분할 수 있다.
- <66>      상기 수신측 다중화/역다중화 제어기 140은 할당되어 있는 물리 채널에서 수신된 정보 비트를 보내면, 물리 채널에서 함께 전달한 FRAME\_QUALITY를 보고서 수신된 정보



비트가 깨졌는지 판단한다. 상기 수신측 다중화/역다중화 제어기 140은 만일 수신된 정보 비트가 깨졌고 수신된 정보 비트가 여러 개의 논리 전송 단위로 나누어져 있다면, 상기 과정에서 구분된 각 논리 전송 단위의 CRC를 다시 평가하여 오류가 없는 논리 전송 단위들이 있는지 살펴본다.

<67>       상기 다중화/역다중화 제어기 140은 상기 과정에서 만일 오류가 있는 논리 전송 단위가 있다면, 각각의 오류가 있는 논리 전송 단위에 대하여 논리적 채널이 연결된 물리 채널에 대응되는 모든 서비스들에게 깨진 데이터 블록이 수신되었음을 알린다.

<68>       상기 수신측 다중화/역다중화 제어기 140은 만일 수신된 정보 비트가 깨졌고 수신된 정보 비트에 하나 또는 여러 개의 MuxPDU 마다 오류를 발견할 수 있는 CRC가 추가되어 있지 않다면, 논리적 채널이 연결된 물리 채널에 대응되는 모든 서비스들에게 깨진 데이터 블록이 수신되었음을 알린다.

<69>       상기 수신측 다중화/역다중화 제어기 140은 만일 오류가 없는 논리 전송 단위 또는 정보 비트가 수신되었으면, 채움 비트열이 아닌 오류가 없는 MuxPDU들을 상기 논리 전송 단위 또는 정보 비트에서 분리한다. 상기 다중화/역다중화 제어기 140은 상기 분리된 MuxPDU가 널 트래픽이나 채움 MuxPDU가 아니라면, 상기 MuxPDU의 서비스 인식자가 지정하는 서비스에게 상기 MuxPDU에 포함된 데이터 블록과 데이터 블록의 길이를 전달한다.

<70>       상기 수신측 다중화/역다중화 제어기 140은 상기 수신 과정을 마치고 나면 오류가 없는 논리 전송 단위 또는 정보 비트가 수신되었고 물리 채널에 논리적 채널이 대응되는 서비스 중에서 데이터 블록을 받지 못한 서비스가 있는 경우, 빈 데이터 블록이 수신되었다고 알린다.

<71>       상기와 같은 다중화/역다중화 제어기 140의 송신 및 수신 동작은 하기와 같은 실시예의 설명을 통해 보다 명확해질 수 있을 것이다. IS-2000 표준에서는 기본 채널(Fundamental Channel), 부가 채널(Supplemental Channel), 전용 제어 채널(Dedicated Control Channel) 등 여러 전용 트래픽 채널(dedicated traffic channel)을 규정하고 있다. 그러므로 본 발명에 따른 다중화/역다중화 제어기 140의 송신 및 수신 동작은 기본 채널에 적용되는 경우와, 부가 채널에 적용되는 경우로 구분되어 설명될 수 있다. 상기 전용 제어 채널은 기본 채널이 9.6 kbps 또는 14.4 kbps로만 동작하는 특수한 경우로 볼 수 있으므로 본 발명에서는 별도의 설명은 하지 않는다. 상기 경우에 덧붙여서 IS-2000 표준에서는 논리 전송 단위가 사용되는 경우와, 논리 전송 단위가 사용되지 않는 경우로 구분되어 설명될 수 있다. 여기서 논리 전송 단위가 사용되는 경우란 콘볼루션 코드(Convolution Code)를 사용하여 데이터를 부호화한 후 데이터를 송수신하는 경우에 해당하며, 논리 전송 단위가 사용되지 않는 경우란 터보코드(Turbo Code)를 사용하여 데이터를 부호화한 후 데이터를 송수신하는 경우에 해당한다.

<72>       먼저, 본 발명의 실시예에 따른 동작을 설명하기 전에 IS-2000 표준에서 규정하고 있는 기본 채널의 정보 비트 수와 부가 채널의 정보 비트 수를 정리해보면 각각 하기의 <표 1> 및 <표 2>, <표 3>, <표 4>와 같다. 하기의 <표 1>과 <표 2>는 IS-2000 표준에서 규정하고 있는 기본 채널의 정보 비트 수를 나타내며, <표 3>과 <표 4>는 부가 채널의 정보 비트 수를 나타낸다. 하기의 <표 1>과 <표 3>은 전송율 9600 bps를 기반으로 하는 전송율 집합 1의 정보 비트 수를 나타내고 하기의 <표 2>와 <표 4>는 전송율 14400 bps를 기반으로 하는 전송율 집합 2의 정보 비트 수를 나타낸다.

## &lt;73&gt; 【표 1】

IS-2000의 기본 채널의 정보 비트 수 (전송율 집합 1)

전송율	정보 비트 수
9600 bps	172 비트
4800 bps	80 비트
2700 bps	40 비트
1500 bps	16 비트

## &lt;74&gt; 【표 2】

IS-2000의 기본 채널의 정보 비트 수 (전송율 집합 2)

전송율	정보 비트 수
14400 bps	267 비트
7200 bps	125 비트
3600 bps	55 비트
1800 bps	21 비트

## &lt;75&gt; 【표 3】

IS-2000의 부가 채널의 정보 비트 수 (전송율 집합 1)

전송율	정보 비트 수
9600 bps	172 비트
19200 bps	360 비트
38400 bps	744 비트
76800 bps	1512 비트
153600 bps	3048 비트
307200 bps	6120 비트
614400 bps	12264 비트

## &lt;76&gt;

【표 4】

IS-2000의 부가 채널의 정보 비트 수 (전송을 집합 2)

전송율	정보 비트 수
14400 bps	267 비트
28800 bps	552 비트
57600 bps	1128 비트
115200 bps	2280 비트
230400 bps	4584 비트
460800 bps	9192 비트
1036800 bps	20712 비트

<77>       상기 <표 1> 및 <표 2>, <표 3>, <표 4>에서는 IS-2000에서 규정하는 모든 정보 비트의 크기를 나타낸 것은 아니라는 사실에 유의하여야 할 것이다.

<78>       상기 <표 3> 또는 <표 4>와 같은 정보 비트 수에 대응하여 LTU를 사용하는 경우, LTU의 크기 및 개수를 구해보면 하기 <표 5>와 <표 6>과 같다. 이때 LTU의 크기와 개수를 곱한 후 남는 비트를 가산하면 정보 비트의 수가 구해진다. 또한 LTU의 페이로드 크기는 하기 <표 5>, <표 6>에서 CRC의 크기인 16 비트를 뺀 값이 된다.

<79>   【표 5】

부가 채널에 적용되는 LTU (전송을 집합 1)

전송율	논리 전송 단위 크기	논리 전송 단위 개수	남는 비트
9600 bps	-	사용하지 않음	-
19200 bps	-	사용하지 않음	-
38400 bps	368 비트	2 개	8 비트
76800 bps	376 비트	4 개	8 비트
153600 bps	376 비트	8 개	40 비트
307200 bps	760 비트	8 개	40 비트
614400 bps	1528 비트	8 개	40 비트

## &lt;80&gt; 【표 6】

부가 채널에 적용되는 LTU (전송율 집합 2)

전송율	논리 전송 단위 크기	논리 전송 단위 개수	남는 비트
14400 bps	-	사용하지 않음	-
28800 bps	-	사용하지 않음	-
57600 bps	560 비트	2 개	8 비트
115200 bps	568 비트	4 개	8 비트
230400 bps	568 비트	8 개	40 비트
460800 bps	1144 비트	8 개	40 비트
1036800 bps	2584 비트	8 개	40 비트

<81>      상기 <표 5>와 <표 6>에서 처음 두 개의 전송율에는 LTU를 사용하지 않는 것에 주의하여야 한다. 즉, 상기 송신측과 수신측 다중화/역다중화 제어기는 이러한 전송율을 가진 부가 채널이 연결되었을 때는 정보 비트를 처리하는 규칙을 따른다.

<82>      정보 비트를 채우기 위해서 본 발명에서 제안하는 MuxPDU 포맷(format)은 하기 <표 7>, <표 8>, <표 9>, <표 10>, <표 11> 및 <표 12>와 같다. 하기 <표 7>과 <표 8>은 기본 채널의 정보 비트들 위한 MuxPDU 포맷을 나타내고, <표 9>와 <표 11>은 LTU가 사용되는 경우에 부가 채널의 정보 비트를 위한 MuxPDU 포맷을 나타내고, <표 10>과 <표 12>는 LTU가 사용되지 않는 경우에 부가 채널의 정보 비트를 위한 MuxPDU 포맷을 나타낸다. 상기한 바와 같이 상기 전용 제어 채널은 기본 채널에서 9600 bps 또는 14400 bps만 허용되는 특수한 경우이므로 하기 <표 7>과 <표 8>에서 전송율 9600 bps 또는 14400 bps에 해당하는 MuxPDU 포맷만 허용되는 것에 주의하여야 한다.

&lt;83&gt;

【표 7】

기본 채널 정보 비트를 위한 MuxPDU 포맷 (전송을 집합 1)

전송율	제1서비스 데이터 블록	시그널링 메시지	서비스 데이터 블록	서비스 인식자	MuxPDU 헤더
9600 bps	171 비트	-	-	-	'0'
	80 비트	88 비트	-	-	'0001'
	40 비트	128 비트	-	-	'0101'
	16 비트	152 비트	-	-	'1001'
	-	168 비트	-	-	'1101'
	80 비트	-	85 비트	3 비트	'0011'
	40 비트	-	125 비트	3 비트	'0111'
	16 비트	-	149 비트	3 비트	'1011'
	-	-	165 비트	3 비트	'1111'
4800 bps	80 비트	-	-	-	-
2700 bps	40 비트	-	-	-	-
1500 bps	16 비트	-	-	-	-

【표 8】

기본 채널 정보 비트를 위한 MuxPDU 포맷 (전송을 집합 2)

전송율	제1서비스 데이터 블록	시그널링 메시지	서비스 데이터 블록	서비스 인식자	MuxPDU 헤더
14400 bps	266 비트	-	-	-	'0'
	124 비트	138 비트	-	-	'00001'
	54 비트	208 비트	-	-	'00011'
	20 비트	242 비트	-	-	'00101'
	-	262 비트	-	-	'00111'
	124 비트	-	135 비트	3 비트	'01001'
	54 비트	-	205 비트	3 비트	'01011'
	20 비트	-	239 비트	3 비트	'01101'
	-	-	259 비트	3 비트	'01111'
	20 비트	222 비트	17 비트	3 비트	'10001'
7200 bps	124 비트	-	-	-	'0'
	54 비트	67 비트	-	-	'0001'
	20 비트	101 비트	-	-	'0011'
	-	121 비트	-	-	'0101'
	54 비트	-	64 비트	3 비트	'0111'
	20 비트	-	98 비트	3 비트	'1001'
	-	-	118 비트	3 비트	'1011'
	20 비트	81 비트	17 비트	3 비트	'1101'
3600 bps	54 비트	-	-	-	'0'
	20 비트	32 비트	-	-	'001'
	-	52 비트	-	-	'011'
	20 비트	-	29 비트	3 비트	'101'
	-	-	49 비트	3 비트	'111'
1800 bps	20 비트	-	-	-	'0'
	-	-	17 비트	3 비트	'1'

<85>      상기 <표 7>과 <표 8>에서 MuxPDU에 포함된 각각의 데이터 블록을 구분할 수 있는 정보가 MuxPDU의 뒷부분에 붙어 있는 점에 주의하여야한다. 상기 MuxPDU 헤더는 각 데이터 블록을 바이트 정렬하기 위해서 MuxPDU의 가장 뒷부분에 위치하였다.

【표 9】

부가 채널 정보 비트를 위한 MuxPDU 포맷 (전송율 집합 1에서 LTU가 사용되는 경우)

전송율	서비스 인식자	길이 지시자	길이 필드	서비스 데이터 블록의 길이
38400 bps	3 비트	'000'	-	최대 346 비트
76800 bps	3 비트	'000'	-	최대 354 비트
153600 bps	3 비트	'000'	-	최대 354 비트
307200 bps	3 비트	'000'	-	최대 738 비트
614400 bps	3 비트	'000'	-	최대 1506 비트
모든 전송율에서	3 비트	'101'	8 비트	최대 2034 비트
모든 전송율에서	3 비트	'110'	16 비트	최대 524266 비트

&lt;87&gt; 【표 10】

부가 채널 정보 비트를 위한 MuxPDU 포맷 (전송율 집합 1에서 LTU가 사용되지 않는 경우)

전송율	서비스 인식자	길이 지시자	길이 필드	서비스 데이터 블록의 길이
19200 bps	3 비트	'000'	-	최대 354 비트
38400 bps	3 비트	'000'	-	최대 738 비트
76800 bps	3 비트	'000'	-	최대 1506 비트
153600 bps	3 비트	'000'	-	최대 3042 비트
153600 bps	3 비트	'100'	8 비트	최대 3034 비트
307200 bps	3 비트	'000'	-	최대 6112 비트
307200 bps	3 비트	'100'	8 비트	최대 6104 비트
614400 bps	3 비트	'000'	-	최대 12258 비트
614400 bps	3 비트	'100'	8 비트	최대 12250 비트
모든 전송율에서	3 비트	'101'	8 비트	최대 2034 비트
모든 전송율에서	3 비트	'110'	16 비트	최대 524266 비트



【표 11】

부가 채널 정보 비트를 위한 MuxPDU 포맷 (전송율 집합 2에서 LTU가 사용되는 경우)

전송율	서비스 인식자	길이 지시자	길이 필드	서비스 데이터 블록의 길이
57600 bps	3 비트	'000'	-	최대 538 비트
115200 bps	3 비트	'000'	-	최대 546 비트
230400 bps	3 비트	'000'	-	최대 546 비트
460800 bps	3 비트	'000'	-	최대 1122 비트
1036800 bps	3 비트	'000'	-	최대 2562 비트
1036800 bps	3 비트	'100'	8 비트	최대 2554 비트
모든 전송율에서	3 비트	'101'	8 비트	최대 2034 비트
모든 전송율에서	3 비트	'110'	16 비트	최대 524266 비트

&lt;89&gt; 【표 12】

부가 채널 정보 비트를 위한 MuxPDU 포맷 (전송율 집합 2에서 LTU가 사용되지 않는 경우)

전송율	서비스 인식자	길이 지시자	길이 필드	서비스 데이터 블록의 길이
28800 bps	3 비트	'000'	-	최대 546 비트
57600 bps	3 비트	'000'	-	최대 1122 비트
115200 bps	3 비트	'000'	-	최대 2274 비트
115200 bps	3 비트	'100'	8 비트	최대 2266 비트
230400 bps	3 비트	'000'	-	최대 4578 비트
230400 bps	3 비트	'100'	8 비트	최대 4570 비트
460800 bps	3 비트	'000'	-	최대 9186 비트
460800 bps	3 비트	'100'	8 비트	최대 9178 비트
1036800 bps	3 비트	'000'	-	최대 20706 비트
1036800 bps	3 비트	'100'	8 비트	최대 20698 비트
모든 전송율에서	3 비트	'101'	8 비트	최대 2034 비트
모든 전송율에서	3 비트	'110'	16 비트	최대 524266 비트

<90>      상기 <표 7>, <표 8>, <표 9>, <표 10>, <표 11>, <표 12>에서 서비스 인식자는  
 다음 <표 13>과 같이 정해질 수 있다.

&lt;91&gt;

【표 13】

서비스 인식자	서비스
서비스 인식자	서비스
'000'	예약
'001'	제1서비스
'010'	제2서비스
'011'	제3서비스
'100'	제4서비스
'101'	제5서비스
'110'	제6서비스
'111'	널 서비스

<92>      상기 <표 13>에서 널 서비스(null service)는 수신측 다중화/역다중화 제어기에게 채우기 위한 채움 MuxPDU임을 알릴 때 사용하는 상기 미리 약속된 특정 서비스 인식자 값이다. 상기 <표 13>에서 알 수 있듯이 상기 <표 7>, <표 8>, <표 9>, <표 10>, <표 11>, <표 12>의 MuxPDU 포맷은 최대 여섯 개의 서비스에서 보내온 데이터 블록을 구분할 수 있다.

<93>      상기 <표 7>과 <표 8>은 기본 채널로 전송되는 MuxPDU 포맷을 보여준다. 여기에서 제1서비스는 서비스 인식자 없이 MuxPDU 헤더만 보고도 구분이 가능하다. 예를 들면, MuxPDU헤더의 하위 2번째 비트가 '0'인 경우가 제1서비스에 해당한다. 제2서비스에서 제6서비스에 해당하는 데이터 블록은 상기 <표 7> 또는 <표 8>의 서비스 인식자 필드를 보고서 판단할 수 있다. 따라서 상기 <표 7> 또는 <표 8>의 서비스 인식자는 '010'에서 '110'까지의 값만을 가질 수 있다. 상기 <표 7> 또는 <표 8>의 MuxPDU 포맷을 사용하는 기본 채널에서는 제1서비스의 데이터 블록이 모두 '1'로 채워지는 경우, 수신측 다중화/역다중화 제어기는 송신측 다중화/역다중화 제어기에서 아무 서비스에게도 해당하지 않는 널 트래픽으로 약속한다. 따라서 수신측 다중화/역다중화 제어기는 기본 채널에서 수

신된 MuxPDU가 제1서비스의 데이터 블록만을 가지며 데이터 블록이 모두 '1'로 채워진 것이면 널 트래픽으로 판단한다.

<94>      상기 <표 7>과 <표 8>에서 전송율 9600 bps 또는 14400 bps에서 허용되는 MuxPDU 포맷은 상기한 바와 같이 전용 제어 채널로 전송 가능하다. 상기한 바와 같이 제2서비스에서 제6서비스에 해당하는 데이터 블록은 상기 <표 7> 또는 <표 8>의 서비스 인식자를 보고서 판단할 수 있다. 상기 전용 제어 채널은 상기한 바와 같이 널 트래픽을 필요로 하지 않는 것에 주의하여야 한다.

<95>      상기 <표 7>과 <표 8>에서 전송율 9600 bps 또는 14400 bps에서 허용되는 MuxPDU 포맷 가운데 시그널링 메시지가 아닌 서비스 데이터 블록을 하나만 포함하는 MuxPDU 포맷은 부가 채널이 9600 bps 또는 14400 bps로 연결되었을 때 전송 가능하다. 상기한 바와 같이 제2서비스에서 제6서비스에 해당하는 데이터 블록은 상기 <표 7> 또는 <표 8>의 서비스 인식자를 보고서 판단할 수 있다.

<96>      상기 <표 9>, <표 10>, <표 11>, <표 12>은 부가 채널이 19200 bps, 28800 bps, 또는 그 이상의 전송율로 연결되었을 때 전송되는 MuxPDU 포맷을 보여준다. 제1서비스에서 제6서비스에 해당하는 데이터 블록은 상기 <표 9>, <표 10>, <표 11>, <표 12>의 서비스 인식자 필드를 보고서 판단할 수 있다. 따라서 상기 <표 9>, <표 10>, <표 11>, <표 12>의 서비스 인식자는 '001'에서 '110'까지의 값을 가질 수 있다. 상기 <표 9>, <표 10>, <표 11>, <표 12>의 MuxPDU 포맷에서 길이 지시자가 '000'인 경우에는 상기 표의 각 전송율에 해당하는 길이만큼의 서비스 블록을 포함한다. 상기 <표 9>, <표 10>, <표 11>, <표 12>의 MuxPDU 포맷에서 길이 지시자가 '100', '101' 또는 '110'인 경우에는 길이 필드를 통하여 MuxPDU의 크기를 계산할 수 있다. 즉, 상기 <표 9>, <표 10>, <표

11>, <표 12>의 MuxPDU 포맷에서 길이 지시자가 '100'인 경우에는 상기 표의 각 전송율에 해당하는 길이에서 길이 필드에 표시된 값에 8을 곱한 만큼을 뺀 길이의 서비스 블록을 가진다. 예를 들어, 전송율 614400 bps로 연결되고 LTU를 사용하지 않는 부가 채널에서 수신된 MuxPDU의 길이 지시자가 '100'이고 길이 필드 값이 '00000010'이면 상기 수신된 MuxPDU에 포함된 서비스 블록의 길이는 상기 <표 9>의 12250 비트에서  $2 \times 8$  비트를 뺀 12234 비트이다. 상기 <표 9>, <표 10>, <표 11>, <표 12>의 MuxPDU 포맷에서 길이 지시자가 '101' 또는 '110'인 경우에는 길이 필드에 표시된 값에 1을 더하고 다시 8을 곱한 값에서 MuxPDU 헤더 크기인 6 비트를 뺀 길이의 서비스 블록을 가진다. 즉, 전송율 614400 bps로 연결되고 LTU를 사용하지 않는 부가 채널에서 수신된 MuxPDU의 길이 지시자가 '110'이고 길이 필드 값이 '0000 0101 1111 1001'이면 상기 수신된 MuxPDU에 포함된 서비스 블록의 길이는 상기 '0000 0101 1111 1001'의 10진수 값인 1529에 1을 더하고 다시 8을 곱한 12240 비트에서 헤더 크기 6 비트를 뺀 12234 비트이다.

#### <97>     다중화/역다중화 제어기의 기본 채널을 통한 송신 동작

<98>     RLP를 사용하는 여섯 개의 서비스가 연결된 경우를 가정하면, 상기 송신측 다중화/역다중화 제어기는 다음과 같이 동작한다. 이러한 동작은 도 10과 같은 흐름에 따른다.

<99>     우선 다중화/역다중화 제어기는 규정된 서비스 품질 보증 규칙(QoS)에 따라서 각각의 서비스들의 전송 순서와 데이터 블록의 크기를 결정한다. 즉 다중화/역다중화 제어기는 시그널링 LAC 계층(LAYER)으로 가능한 크기를 문의하고(도 10의 S10단계), 상기 시그

날링 LAC 계층으로부터 적당한 크기의 데이터 블록을 확인한다(S11단계). 다음에, 상기 다중화/역다중화 제어기는 서비스 전송순서를 결정하고(S12단계), 다음에 상기 확인된 크기의 데이터 블록을 우선적으로 제1서비스에 요청하고(S13단계), 상기 제1서비스로부터 상기 확인된 크기내의 데이터 블록을 전달받는다(S14단계). 기본 채널로 전송될 데이터 블록의 경우, 상기 <표 7> 또는 <표 8>에서 MuxPDU가 허용하는 데이터 블록의 크기와 개수, 그리고 조합에 따라서 적절한 크기의 데이터 블록을 생성하도록 RLP처리기에게 요청하여야 한다. 그런 다음에, 상기 다중화/역다중화 제어기는 전송할 데이터 블록을 누적하고, 전송할 수 있는 남아있는 블록을 계산한다(S15단계). 다음에, 상기 다중화/역다중화 제어기는 상기 누적한 데이터 블록으로 MuxPDU를 만들 수 있는지 여부를 판단한다(S16단계). MuxPDU를 만들 수 없는 경우에는 다시 S13단계로 진행하여 해당하는 서비스로 데이터 블록을 요청하고, 그 요청에 따른 데이터 블록을 전달받는다. MuxPDU를 만들 수 있는 경우에는 누적된 데이터 블록으로 MuxPDU를 만든다(S17단계). 이때 다중화/역다중화 제어기는 포함되는 데이터 블록에 따라서 상기 <표 4>로부터 적절한 비트열을 찾아 MuxPDU 헤더에 덧붙인다. 상기 다중화/역다중화 제어기는 이렇게 생성된 MuxPDU를 정보 비트로 하여 상기 물리 채널에게 전달한다(S18단계).

<100>      상기 과정에서 데이터 블록을 생성할 기회를 얻지 못한 RLP처리기에게는 상기 다중화/역다중화 제어기가 빈 데이터 블록(blank data block)을 생성하도록 요청하여 상기 RLP처리기가 전송할 기회를 얻지 못하였음을 알 수 있도록 한다. 또한 상기 과정에서 모든 RLP가 아무런 데이터 블록도 제공하지 않았다면, 상기 다중화/역다중화 제어기는 상기 널 트래픽을 만들어서 이것을 정보 비트로 하여 상기 물리 채널에게 전달한다.

<101>      다중화/역다중화 제어기의 기본 채널을 통한 수신 동작

<102>      수신측 다중화/역다중화 제어기는 기본 채널로 전송된 정보 비트에 대해서 다음과 같이 동작한다. 이러한 동작은 도 11과 같은 처리흐름에 따른다.

<103>      우선 수신된 정보 비트의 전송율과 MuxPDU 헤더 비트를 보고서(도 11의 S20단계) 각각의 데이터 블록을 구분한다(S21, S22단계). 각각의 데이터 블록을 구분하기 위해서는 먼저 전송율 집합에 따라서 상기 <표 7> 또는 <표 8>을 참조한다. 우선 수신된 정보 비트의 마지막 1 비트를 보고 이 값이 '0'으로 설정되어 있으면, 마지막 1 비트를 제외한 나머지 모든 정보 비트는 제1서비스의 데이터 블록이므로 제1서비스에게 데이터 블록의 길이와 함께 전달한다(S23단계).

<104>      상기 수신측 다중화/역다중화 제어기는 상기 과정에서 만일 수신된 정보 비트의 마지막 1 비트가 '1'로 설정되어 있으면 전송율 집합 1의 경우 마지막 4 비트를, 전송율 집합 2의 경우 마지막 5 비트를 MuxPDU 헤더로 간주한다. 상기 MuxPDU 헤더와 같은 비트열을 가지는 조합을 상기 <표 7> 또는 <표 8>에서 찾는다. 만일 같은 비트열을 가지는 조합이 없으면 상기 수신측 다중화/역다중화 제어기는 수신된 정보 비트를 깨진 것으로 처리한다. 만일 같은 비트열을 가지는 조합이 있으면 상기 <표 7> 또는 <표 8>에 규정된 데이터 블록의 크기와 위치에 따라서 데이터 블록과 서비스 인식자를 구분한다. 예를 들어 전송율 집합 1일 때 수신된 정보 비트가 9600 bps로 수신되었고 상기 MuxPDU 헤더가 '0011'이라면 수신된 MuxPDU는 첫 80 비트는 제1서비스 데이터 블록이고 그 다음 85 비트는 또다른 서비스의 데이터 블록이며 나머지 3 비트는 서비스 인식자가 된다. 상기 <표 7>

> 또는 <표 8>에 규정된 것과 같이 조합에 따라서는 서비스 인식자가 없을 수도 있다.

<105>      상기 과정에서 구분된 데이터 블록은 상기 <표 7> 또는 <표 8>을 참조하여 해당하는 서비스에게 전달한다. 제1서비스와 시그널링 계층에는 서비스 인식자를 보지 않고도 바로 데이터 블록과 데이터 블록의 길이를 함께 전달할 수 있다. 그러나 서비스 인식자가 있는 경우에는 서비스 인식자를 상기 <표 13>과 대조하여 해당하는 서비스에게 데이터 블록과 데이터 블록의 길이를 함께 전달한다. 상기 예에서 수신측 다중화/역다중화 제어기는 상기 80 비트의 데이터 블록을 그 길이와 함께 제1서비스에게 전달하고 서비스 인식자가 가리키는 서비스에게 상기 85 비트의 데이터 블록을 그 길이와 함께 전달한다. 그러나 서비스 인식자의 값이 '000', '001', '111'로 되어 있으면 상기 수신측 다중화/역다중화 제어기는 상기 수신된 정보 비트를 깨진 것으로 처리한다.

<106>      만일 수신된 정보 비트가 깨진 것이라면, 상기 다중화/역다중화 제어기는 기본 채널에 논리적 채널이 대응되는 모든 서비스에게 깨진 데이터 블록이 수신되었음을 알린다. 이와 달리, 상기 수신측 다중화/역다중화 제어기는 상기 정보 비트가 깨지지 않았고 데이터 블록이 하나 밖에 없으며 제1서비스에 해당하는 데이터 블록이고 모두 '1'로 채워져 있으면, 널 트래픽으로 간주하고 버리고 기본 채널에 논리적 채널이 모든 대응되는 서비스들에게 아무런 데이터 블록이 수신되지 않았음을 알린다.

<107>      상기 수신측 다중화/역다중화 제어기는 상기 정보 비트가 깨지지 않았을 때 기본 채널에 논리적 채널이 대응되는 서비스 중에서 수신된 데이터 블록이 없는 서비스들에 대해서 빈 데이터 블록(blank data block)이 수신되었음을 알려준다. 상기 널 트래픽인 경우 아무런 데이터 블록이 수신되지 않았음을 알리는 것과 상기 빈 데이터 블록이 수신되었음을 알리는 것은 서비스에 따라서 다른 의미를 가질 수 있음에 주의하여야한다.

<108>      다중화/역다중화 제어기의 부가 채널을 통한 송신 동작

<109>      다중화/역다중화 제어기가 부가 채널 정보 비트를 생성할 때 LTU를 생성하는 경우 전송율에 따라 <표 5> 또는 <표 6>에 표시된 개수만큼의 LTU를 만든다. 상기 LTU는 <표 5> 또는 <표 6>에 규정된 것과 같은 크기를 갖는다. 상기 LTU는 16비트 CRC를 가지므로 실제 LTU에 실어 보낼 수 있는 최대 MuxPDU의 크기는 전송율에 따라 <표 5> 또는 <표 6>에 표시된 크기에서 16 비트를 뺀 값이 된다. 예를 들어, 부가 채널 307.2 kbps를 사용하고 LTU를 생성하는 경우에는 LTU의 페이로드에 MuxPDU가 들어가게 되므로 MuxPDU의 최대 크기는 LTU 페이로드의 크기 760 비트에서 CRC 16 비트를 뺀 나머지 744 비트가 된다. 상기 다중화/역다중화 제어기가 부가 채널 정보 비트를 생성할 때는 LTU를 생성하는 경우 가능한 MuxPDU 포맷은 전송율 집합에 따라 상기 <표 9>와 <표 11>의 포맷이 가능하다. 상기 다중화/역다중화 제어기가 MuxPDU를 생성하여 LTU 페이로드를 모두 채우면 상기 다중화/역다중화 제어기는 LTU 페이로드에 대한 16 비트 CRC를 생성한다. 상기 16 비트 CRC는 부가 채널에 적용되는 16 비트 CRC 생성 방식과 똑같이 만든다. 상기 다중화/역다중화 제어기는 상기 방식으로 상기 <표 5> 또는 <표 6>에 규정된 것과 같은 개수의 LTU를 만들고 이를 차례로 정보 비트에 넣은 후 남은 공간은 '0'으로 채우고 물리 계층 처리기에게 전달한다.

<110>      상기 다중화/역다중화 제어기가 부가 채널 정보 비트를 생성할 때 LTU를 생성하지 않는 경우 전송율에 따라 <표 3> 또는 <표 4>에 지정된 크기의 정보 비트를 생성한다. 이러한 경우 정보 비트에 실어 보낼 수 있는 MuxPDU는 전송율 9600 bps 또는 14400 bps



인 경우 상기한 바와 같이 상기 <표 7>과 <표 8>에 규정된 서비스 데이터 블록이 하나만 들어간 하나의 MuxPDU만 전송이 가능하다. 상기 다중화/역다중화 제어기가 MuxPDU를 생성하여 정보 비트를 채우면 이를 물리 계층 처리기에게 전달한다.

<111>      상기 다중화/역다중화 제어기가 부가 채널 정보 비트를 생성할 때 LTU를 생성하지 않는 경우 전송율이 19200 bps 또는 28800 bps 이상일 때에는 상기 <표 10>과 <표 12> MuxPDU 포맷이 가능하다. 상기 다중화/역다중화 제어기가 MuxPDU를 생성하여 정보 비트를 채우면 이를 물리 계층 처리기에게 전달한다.

<112>      상기 다중화/역다중화 제어기는 규정된 서비스 품질 보증 규칙(QoS)에 따라서 각각의 서비스들의 전송 순서와 데이터 블록의 크기를 결정한다. 다음에, 상기 다중화/역다중화 제어기는 우선 순위에 따라서 각 서비스들의 RLP에게 데이터 블록을 요청한다(도 12의 S30단계). 즉, 다중화/역다중화 제어기는 우선순위가 가장 높은 제1서비스의 RLP처리기로 가능한 데이터 블록을 요청하고(S30단계), 상기 제1서비스의 RLP처리기로부터 대응하는 데이터 블록 또는 빈데이터 블록을 전달받는다(S31단계). 이때 빈데이터 블록이 아닌 데이터블록이 전달되는 경우 다중화/역다중화 제어기는 이 전달된 데이터블록을 이용하여 MuxPDU를 생성한다(S32단계). 이렇게 생성된 MuxPDU는 결과적으로 정보비트로서 생성되게 된다.

<113>      상기 다중화/역다중화 제어기가 부가 채널 정보 비트를 생성할 때 LTU를 생성하는 경우, 상기 <표 9> 또는 <표 11>에서 MuxPDU가 허용하는 데이터 블록의 크기와 현재 생성 중인 LTU의 남은 자리에 따라서 적절한 크기의 데이터 블록을 생성하도록 RLP처리기에게 요청하여야 한다. 즉, 다중화/역다중화 제어기는 데이터 블록을 전달받은 후 LTU

또는 정보 비트의 남은 크기를 계산한 후(S33단계), 이때 계산된 크기가 가능한 MuxPDU의 크기 이상인지 여부를 판단한다(S34단계). 상기 계산된 크기가 가능한 MuxPDU의 크기 이상인 것으로 판단되는 경우, 다중화/역다중화 제어기는 가능한 크기의 데이터 블록을 다음 우선순위의 제2서비스의 RLP처리기로 요청하고(S35단계), 상기 제2서비스의 RLP처리기로부터 대응하는 데이터 블록 또는 빈 데이터 블록을 전달받는다(S36단계). 위와 같은 동작은 모든 서비스의 RLP처리기에 대해 반복적으로 수행될 수 있다.

<114>      상기 다중화/역다중화 제어기가 부가 채널 정보 비트를 생성할 때 LTU를 생성하지 않고 전송율 9600 bps 또는 14400 bps인 경우에는, 상기 <표 7>과 <표 8>에 규정된 MuxPDU 포맷 중에서 부가 채널로 보낼 수 있는 MuxPDU를 하나 생성하기 위하여 우선 순위에 따라 각각의 서비스에게 상기 <표 7>과 <표 8>에 규정된 부가 채널로 보낼 수 있는 데이터 블록을 생성하도록 요청한다. 만일 어느 한 서비스라도 데이터 블록을 생성하면 상기 다중화/역다중화 제어기는 이를 MuxPDU로 만든다. 상기 다중화/역다중화 제어기가 부가 채널 정보 비트를 생성할 때 LTU를 생성하지 않는 경우 전송율이 19200 bps 또는 28800 bps 이상인 경우에는, 상기 <표 10> 또는 <표 12>에서 MuxPDU가 허용하는 데이터 블록의 크기와 현재 생성 중인 LTU의 남은 자리에 따라서 적절한 크기의 데이터 블록을 생성하도록 RLP처리기에 요청하여야 한다(도 12의 S32단계~S38단계).

<115>      상기 다중화/역다중화 제어기가 부가 채널 정보 비트를 생성할 때 LTU 페이로드를 채우거나 정보 비트를 채우는 경우 상기 다중화/역다중화 제어기는 현재까지 채우지 못한 LTU 페이로드 또는 정보 비트의 나머지 부분의 길이를 알고 있어야 한다. 상기 다중화/역다중화 제어기는 남은 부분에 대해서 서비스들에게 데이터 블록을 생성하도록 요청한다. 만일 어떤 서비스가 빈 데이터 블록이 아닌 데이터 블록을 전달하면 상기 다중화/

역다중화 제어기는 이 데이터 블록의 길이에 따라서 다음과 같이 동작한다.

<116> 첫째, 만일 전달된 데이터 블록의 길이가 LTU 페이로드 또는 정보 비트의 남은 공간에 비해서 6 비트만큼 작으면 상기 다중화/역다중화 제어기는 전송한 서비스에 따라서 <표 13>을 참고하여 3 비트 서비스 인식자를 붙이고 길이 지시자 3 비트를 '000'으로 설정하여 상기 데이터 블록의 앞에 붙여서 MuxPDU를 만든다. 상기 다중화/역다중화 제어기는 상기 생성된 MuxPDU를 LTU 페이로드 또는 정보 비트의 남은 공간에 넣고 LTU 페이로드 또는 정보 비트를 완성시킨다.

<117> 둘째, 만일 전달된 데이터 블록의 길이가 LTU 페이로드 또는 정보 비트의 남은 공간에 비해서 14 비트 또는 그 이상 작으면 상기 다중화/역다중화 제어기는 상기 <표 9>, <표 10>, <표 11>, <표 12>에 제시된 8 비트 또는 16 비트 길이 필드를 가지는 MuxPDU를 생성한다. 즉, 상기 다중화/역다중화 제어기는 생성된 데이터 블록이 2034 비트와 같거나 더 작으면 전송한 서비스에 따라서 <표 13>을 참고하여 3 비트 서비스 인식자를 붙이고 길이 지시자 1 비트를 '101'로, 그리고 길이 필드 8 비트를 상기 서비스 인식자, 길이 지시자, 길이 타입 필드, 길이 필드, 그리고 데이터 블록을 합한 전체 MuxPDU의 길이를 바이트 단위로 표시하여 이 값에서 1을 뺀 값으로 설정한다. 만일 생성된 데이터 블록이 2034 비트 보다 더 크면 전송한 서비스에 따라서 <표 13>을 참고하여 3 비트 서비스 인식자를 붙이고 길이 지시자 1 비트를 '110'으로, 그리고 길이 필드 16 비트를 상기 서비스 인식자, 길이 지시자, 길이 타입 필드, 길이 필드, 그리고 데이터 블록을 합한 전체 MuxPDU의 길이를 바이트 단위로 표시하여 이 값에서 1을 뺀 값으로 설정한다. 상기 다중화/역다중화 제어기는 상기 <표 9>, <표 10>, <표 11>, <표 12>에 제시된 8 비트 길이 필드를 가지고 길이 지시자를 '100'으로 가지는 MuxPDU를 생성할 수도

있다. 즉, <표 9>, <표 10>, <표 11>, <표 12>에서 제시된 최대 데이터 블록의 크기에서 얼마큼 모자라는지를 바이트 단위로 표시한 8 비트 길이 필드를 넣어서 MuxPDU를 생성할 수 있다. 상기 다중화/역다중화 제어기는 만일 상기 과정에서 생성된 MuxPDU의 크기가 정수가 아니면, 즉 MuxPDU의 길이가 바이트 단위가 아니면 상기 다중화/역다중화 제어기는 상기 데이터 블록을 버린다. 만일 생성된 MuxPDU의 크기가 정수이면 상기 서비스 인식자, 길이 지시자, 길이 타입 필드, 길이 필드를 상기 데이터 블록의 앞에 붙여서 MuxPDU를 만든다. 상기 다중화/역다중화 제어기는 상기 생성된 MuxPDU를 LTU 페이로드 또는 정보 비트의 남은 공간에 넣는다.

<118>       상기와 같이 생성된 MuxPDU를 LTU의 페이로드(payload)에 차례로 집어 넣고 남은 부분에 대해서 다시 상기 과정을 반복한다. 이때 적절한 크기의 데이터 블록이 더 이상 없다면, 상기 다중화/역다중화 제어기는 서비스 인식자를 '111', 그리고 길이 지시자를 '000'으로 하여 남은 부분의 첫번째 6 비트를 채운 후 나머지 부분을 모두 '0'으로 설정하여 LTU 페이로드 또는 정보 비트를 채운다. LTU를 생성하는 경우 상기 과정을 통하여 <표 5> 또는 <표 6>에 제시된 필요한 개수의 LTU를 생성하면, 다중화/역다중화 제어기는 생성된 LTU를 모두 정보 비트에 차례로 집어넣는다. 상기 다중화/역다중화 제어기는 상기 <표 5> 또는 <표 6>에 나타난 것과 같이 남은 비트 부분은 모두 '0'으로 채우고 이를 물리 채널 제어기에 전달한다.

<119>       LTU를 생성하지 않는 경우 상기 과정을 통하여 <표 3> 또는 <표 4>에 제시된 정보 비트를 모두 채우면 상기 다중화/역다중화 제어기는 이를 물리 채널 제어기에 전달한다.

<120>      상기 과정을 통하여 얻을 수 있는 정보 비트의 예가 도 6에 나타나 있다. 상기 도 6에서는 세 개의 LTU를 보여준다. 도 6의 (a)를 참조하면, 첫번째 LTU는 제1서비스에서 738 비트의 데이터 블록을 받은 경우로 LTU 페이로드 보다 정확히 6 비트 작으므로, 서비스 인식자를 제1서비스 '001'로 설정하고 길이 지시자를 '000'으로 설정한 다음 받은 데이터 블록으로 LTU의 페이로드를 채운다. 도 6의 (b)를 참조하면, 두번째 LTU는 제2서비스에서 330 비트의 데이터 블록을 받은 경우로 남은 LTU 페이로드 보다 14 비트 이상 작고 2034 비트 보다는 작으므로, 서비스 인식자를 제2서비스 '010'으로 설정하고 길이 지시자를 '101'로 설정한 다음 길이 필드를 MuxPDU 전체 길이인 43 바이트에서 하나 뺀 값인 '0010 1010'으로 설정한다. 나머지 50 바이트의 LTU의 페이로드 부분은 아무런 서비스도 데이터 블록을 제공하지 않은 경우로 이때에는 채움 MuxPDU를 생성하여 집어 넣는다. 도 6의 (c)를 참조하면, 세번째 LTU는 LTU를 생성할 때 아무런 서비스도 데이터 블록을 제공하지 않은 경우로, 역시 채움 MuxPDU를 생성하여 집어 넣는다. 상기 도 6에서 예로 제시된 LTU들을 모아서 정보 비트를 채우고 남은 비트를 '000'으로 설정하면 정보 비트의 생성이 끝나게 된다.

<121>      다중화/역다중화 제어기의 부가채널을 통한 수신 동작

<122>      수신측 다중화/역다중화 제어기는 부가 채널로 전송된 정보 비트에 대해서 다음과 같이 동작한다. 이러한 동작은 도 13에 도시된 바와 같은 흐름에 따른다.

<123> LTU가 사용된 정보 비트의 경우, 상기 <표 5> 또는 <표 6>에서와 같이 전송율에 따라 LTU를 구분한다. 예를 들어, 307.2 kbps로 연결된 부가 채널로 수신된 정보 비트는 상기 <표 5>에서와 같이 760 비트 단위로 LTU를 구분한다. 만일 정보 비트가 오류가 없으면, 각각의 LTU 또는 정보 비트에 대해서 MuxPDU를 구분하는 작업을 한다(도 13의 S40 단계). MuxPDU를 구분한 이후에 다중화/역다중화 제어기는 데이터 블록이 전달될 서비스 RLP를 판단하고(S41단계), 해당하는 서비스의 RLP로 수신 데이터 블록을 전달한다. 이때 수신 데이터 블록과 함께 데이터 블록의 길이를 전달한다(S42, S43단계). 이러한 수신 데이터 블록과 데이터 블록의 길이의 전달은 구분된 모든 MuxPDU에 대해 해당하는 서비스의 RLP로 이루어진다.

<124> 이와 달리, 정보 비트가 오류가 있으면, 각각의 LTU에 대해서 CRC를 검사한다. 이때 CRC 검사에서 오류가 없는 LTU에 대해서는 MuxPDU를 구분한다. 각각의 오류가 있는 LTU에 대해서는 부가 채널에 논리적 채널이 대응되는 모든 서비스에게 깨진 데이터 블록이 수신되었음을 알리고, 각 서비스가 상기 깨진 LTU로 보낼 수 있는 최대 데이터 블록의 길이를 함께 알려준 다음, 상기 정보 비트를 버린다. 상기 예에서 307.2 kbps로 연결된 부가 채널로 수신된 상기 LTU로 보낼 수 있는 최대 데이터 블록의 길이는 상기 도 6에서와 같이 740 비트가 될 것이다.

<125> LTU가 사용되지 않은 정보 비트의 경우, 전송율이 9600 bps 또는 14400 bps인 경우에는 상기<표 7> 또는 <표 8>에 따라서 MuxPDU를 구분한다. 이때 MuxPDU를 구분하는 방법은 기본 채널에서와 같이 수행하면 된다. 단 부가 채널에서는 부가 채널에 논리적 채널이 대응되는 서비스에서 받은 하나의 데이터 블록만 MuxPDU에 있을 수 있으므로 MuxPDU 헤더가 이와 같지 않은 정보 비트는 깨진 것으로 간주한다.

- <126> LTU가 사용되지 않은 정보 비트의 경우, 전송율이 19200 bps 또는 28800 bps 이상인 경우에는 전체 정보 비트에 대해서 MuxPDU를 구분한다. 만일 정보 비트가 오류가 있으면, 상기 다중화/역다중화 제어기는 부가 채널에 논리적 채널이 대응되는 모든 서비스에게 깨진 데이터 블록이 수신되었음을 알려준 다음, 상기 정보 비트를 버린다.
- <127> 상기 LTU 페이로드 또는 정보 비트에 대해서 MuxPDU를 구분하는 경우, 상기 서비스 인식자와, 길이 지시자, 그리고 길이 필드를 통하여 MuxPDU가 가지고 있는 데이터 블록이 어떤 서비스에게 전달되어야 하는지와 수신된 MuxPDU의 전체 길이를 다음과 같이 알 수 있다.
- <128> 첫째, 상기 수신측 다중화/역다중화 제어기는 상기 LTU 페이로드 또는 정보 비트의 가장 앞 부분에서부터 MuxPDU 구분을 시작한다.
- <129> 둘째, 해석 중인 MuxPDU의 서비스 인식자가 상기 <표 13>에서와 같이 '111'로 되어 있으면 상기 수신측 다중화/역다중화 제어기는 상기 LTU 페이로드 또는 정보 비트의 남은 부분을 모두 무시한다.
- <130> 셋째, 만일 해석 중인 MuxPDU의 서비스 인식자가 상기 <표 13>에서와 같이 '000'로 되어 있으면 상기 수신측 다중화/역다중화 제어기는 상기 LTU 페이로드 또는 정보 비트를 깨진 것으로 간주한다.
- <131> 넷째, 만일 해석 중인 MuxPDU의 서비스 인식자가 상기 <표 13>에서와 같이 '001'에서 '110' 사이이고 부가 채널에 논리적 채널이 대응되는 서비스 중 하나를 가리키면 다음 3 비트 길이 지시자를 본다. 만일 서비스 인식자가 부가 채널에 논리적 채널이 대응되는 서비스 중 하나를 가리키지 않으면 상기 수신측 다중화/역다중화 제어기는 상기

LTU 페이로드 또는 정보 비트를 깨진 것으로 간주한다.

- <132> 다섯째, 만일 3 비트 길이 지시자가 '000'이면 상기 LTU 페이로드 또는 정보 비트의 남은 부분이 모두 하나의 MuxPDU이다. 따라서 남은 부분의 크기에서 6 비트를 뺀 값을 길이로 하는 MuxPDU에서 데이터 블록을 상위 서비스에게 전달한다.
- <133> 여섯째, 만일 3 비트 길이 지시자가 '100', '101', 또는 '110'이면 다음의 길이 필드를 본다. 만일 3 비트 길이 지시자가 '000', '100', '101', 또는 '110'이 아니면 상기 수신측 다중화/역다중화 제어기는 상기 LTU 페이로드 또는 정보 비트를 깨진 것으로 간주한다.
- <134> 일곱째, 만일 3 비트 길이 타입 필드가 '100'이면 상기 LTU 페이로드 또는 정보 비트에 들어갈 수 있는 최대 크기의 MuxPDU에서 8 비트 길이 필드 값에 8을 곱한 만큼을 뺀 길이에 해당하는 부분이 하나의 MuxPDU이다. 따라서 상기 길이의 MuxPDU에서 앞 14 비트 헤더 부분을 제외한 나머지 부분을 데이터 블록으로 하여 데이터 블록의 길이와 함께 상위 서비스에게 전달한다.
- <135> 여덟째, 만일 3 비트 길이 타입 필드가 '101' 또는 '110'이면 다음 8 비트 또는 16 비트의 길이 필드를 본다. 만일 길이 비트의 값에 1을 더한 값이 상기 LTU 페이로드 또는 정보 비트의 남은 부분의 길이를 바이트로 표시한 것 보다 크면 상기 수신측 다중화/역다중화 제어기는 상기 LTU 페이로드 또는 정보 비트를 깨진 것으로 간주한다.
- <136> 아홉째, 만일 상기 길이 필드 값에 1을 더한 값이 상기 LTU 페이로드 또는 정보 비트의 남은 부분의 길이를 바이트로 표시한 것과 같거나 작으면 상기 구해진 값이 MuxPDU의 길이이다. 따라서 상기 LTU 페이로드 또는 정보 비트의 남은 부분에서 상기 구해진



값을 길이로 하는 MuxPDU에서 앞 14 비트 또는 22 비트의 헤더 부분을 뺀 나머지 부분을 데이터 블록으로 하여 데이터 블록의 길이와 함께 상위 서비스에게 전달한다.

<137> 마지막으로, 상기 MuxPDU를 구하고 상기 LTU 페이로드 또는 정보 비트의 남은 부분이 있으면 남은 부분에 대해서 상기 과정을 다시 수행한다.

<138> 상기 도 3 및 도 4에 도시된 바와 같은 RLP제어기 131에 의해 수행되는 본 발명에 따른 다음과 같이 구분되어 설명될 수 있다.

<139> RLP제어기의 데이터 송신 동작

<140> RLP제어기 131은 상기 송신 데이터 버퍼 122에 저장된 데이터열(data stream)을 받아서 상기 다중화/역다중화 제어기 140이 데이터 블록을 요구할 때마다 요구된 데이터 블록의 크기에 맞는 RLP 프레임을 생성하여 전달한다. 상기 송신 데이터 버퍼 122에 저장된 데이터열은 무한한 길이를 가지는 것으로 간주된다. 따라서 RLP제어기 131은 이러한 데이터열의 시작 부분과 끝 부분을 특별히 구별하여 전송하지 않는다.

<141> RLP제어기 131은 상기 송신 데이터 버퍼 122에 저장된 데이터열을 적절한 길이의 블록(block)으로 나눈다. 상기 블록들은 모두 같은 크기를 가질 수도 있고 각각 다른 크기를 가질 수도 있다. 그러나 상기 블록은 모두 크기가 한 바이트 이상이고 최대  $N_{DATA}$  바이트까지 가능하다.

<142> RLP제어기 131은 상기 블록에게 0에서  $(N_{BLOCK} - 1)$ 까지의 일련 번호를 부여한

다. 본 발명에서는 상기 블록에게 부여되는 일련 번호를 '블록 일련 번호(block sequence number)'라고 칭하겠다. 상기 RLP제어기 131은 가장 먼저 만들어지는 블록에게 일련 번호 0번을 부여한다. 상기 RLP제어기 131은 새로운 블록을 만들면, 앞의 블록의 일련 번호에서 하나 증가된 일련 번호를 부여한다. 만일 앞의 블록이 ( $N_{BLOCK} - 1$ )의 일련 번호를 받았다면 상기 RLP제어기 131은 새로운 블록에게 다시 일련 번호 0번을 부여한다.

<143> RLP제어기 131은 또한 새로운 블록의 데이터 바이트 각각에 대해서 0에서 ( $N_{DATA} - 1$ )까지의 일련 번호를 부여한다. 본 발명에서는 상기 데이터 바이트에게 부여되는 일련 번호를 '데이터 일련 번호(data sequence number)'라고 칭하겠다. 상기 RLP제어기 131은 블록의 첫번째 데이터 바이트에게 일련 번호 0번을 부여한다. 상기 RLP제어기 131은 다음번 데이터 바이트에게 앞의 데이터 바이트의 일련 번호에서 하나 증가된 일련 번호를 부여한다. 상기한 바와 같이 각 블록은 최대  $N_{DATA}$  바이트까지 데이터 바이트를 가질 수 있으므로, 0에서 ( $N_{DATA} - 1$ )까지의 일련 번호는 데이터 일련 번호로 충분히 사용할 수 있다.

<144> 상기 블록 일련 번호와 데이터 일련 번호를 가지면 상기 데이터열에서 특정한 데이터 바이트를 지정할 수 있게 된다. 상기 블록 일련 번호와 데이터 일련 번호를 가지고 특정한 데이터 바이트를 지정할 수 있는 방식을 본 발명에서는 '2차원 주소 방식(two-dimensional addressing)'이라고 칭하겠다.

<145> RLP제어기 131은 상기 다중화/역다중화 제어기 140이 데이터 블록을 요청하면 첫번째 블록의 첫번째 바이트부터 적절한 크기의 데이터를 사용하여 RLP 프레임을 만든다. 이때 상기 RLP제어기 131은 수신된 RLP 프레임이 담아온 데이터 조각이 어떤 블록의 몇

번째 데이터부터인지 수신측 RLP제어기가 알 수 있도록 해당하는 데이터 조각의 블록 일련 번호와 데이터 일련 번호를 RLP 프레임에 포함시켜 전송한다.

<146> RLP제어기 131은 상기 RLP 프레임을 만들 때 전송하는 블록의 크기가  $N_{DATA}$  바이트보다 작은 경우, 수신측 RLP제어기에서 상기 전송된 블록의 끝을 알 수 있도록 해야한다. 즉, 상기 RLP제어기 131은 전송하는 블록이  $N_{DATA}$  바이트보다 작은 데이터를 가지고 있을 경우, 수신측 RLP제어기에게 블록의 마지막을 알려서 수신측 RLP제어기가 다음번 블록을 기다리도록 한다. 본 발명에서는 이러한 목적으로 '블록 끝 지시자(block end indicator)'를 제안한다. 즉, 상기 송신측 RLP제어기는 상기 RLP 프레임을 만들 때 전송하는 블록의 크기가  $N_{DATA}$  바이트보다 작은 경우 블록 끝 지시자(block end indicator)를 함께 보낸다. 상기 RLP제어기 131은 1 비트의 블록 끝 지시자를 크기가  $N_{DATA}$  바이트보다 작은 블록의 마지막 부분을 보낼 때 '1'로 설정하고 그렇지 않을 때는 '0'으로 설정한다. 상기 블록 끝 지시자가 '1'로 설정된 RLP 프레임을 받으면 수신측 RLP제어기는 다음 일련 번호를 갖는 블록의 첫번째 데이터 바이트가 도착하기를 기다리게 된다.

<147> 상기 과정의 예가 도 7에 도시되어 있다. 상기 도 7에서  $N_{DATA}$ 는 6으로 가정된다. 상기 도 7에서 RLP 제어기 131은 상기 다중화/역다중화 제어기 140의 요청에 따라서 모두 5개의 데이터 블록, 즉 5개의 RLP 프레임을 만든다. 상기 다중화/역다중화 제어기 140에서 6 바이트의 데이터를 담을 수 있는 데이터 블록을 요청하면, 상기 RLP제어기 131은 상기 도 7의 RLP 프레임 A와 같이 첫번째 RLP 프레임을 생성한다. 상기 도 7에서 첫번째 블록인 일련 번호 0번 블록이 총 6 바이트의 데이터를 가지고 있으므로, 상기 RLP제어기 131은 상기 0번 블록만으로 RLP 프레임을 만들 수 있다. 상기 RLP제어기 131은 상기 일련 번호 0번 블록의 일련 번호 0번 데이터 바이트부터 시작된 데이터 조각으

로 RLP 프레임을 만들었으므로, 블록 일련 번호 0 그리고 데이터 일련 번호 0을 RLP 프레임 헤더에 표시한다. 상기 RLP제어기 131은 또한 상기 일련 번호 0번 블록의 크기가 6 바이트 이므로 블록 끝 지시자를 '0'으로 설정한다.

<148>      상기 다중화/역다중화 제어기 140에서 4 바이트의 데이터를 담을 수 있는 데이터 블록을 요청하면, 상기 RLP제어기 131은 상기 도 7의 RLP 프레임 B와 같이 두번째 RLP 프레임을 생성한다. 상기 도 7에서 첫번째 블록은 이미 모두 전송되었으므로, 상기 RLP 제어기 131은 두번째 블록인 일련 번호 1번 블록의 데이터를 사용하여 RLP 프레임을 생성한다. 상기 도 7에서 일련 번호 1번 블록이 4 바이트의 데이터를 가지고 있으므로, 상기 RLP제어기 131은 상기 1번 블록만으로 RLP 프레임을 만들 수 있다. 상기 RLP제어기 131은 상기 일련 번호 1번 블록의 일련 번호 0번 데이터 바이트부터 시작된 데이터 조각으로 RLP 프레임을 만들었으므로, 블록 일련 번호 1 그리고 데이터 일련 번호 0을 RLP 프레임 헤더에 표시한다. 상기 RLP제어기 131은 또한 상기 일련 번호 1번 블록의 크기가 6 바이트 보다 작은 4 바이트이고 1번 블록의 마지막 부분을 담아서 보내므로 블록 끝 지시자를 '1'로 설정한다.

<149>      상기 다중화/역다중화 제어기 140에서 다시 4 바이트의 데이터를 담을 수 있는 데이터 블록을 요청하면, 상기 RLP제어기 131은 상기 도 7의 RLP 프레임 C와 같이 세번째 RLP 프레임을 생성한다. 상기 도 7에서 첫번째와 두번째 블록은 이미 모두 전송되었으므로, 상기 RLP제어기 131은 세번째 블록인 일련 번호 2번 블록의 데이터를 사용하여 RLP 프레임을 생성한다. 상기 도 7에서 일련 번호 2번 블록이 6 바이트의 데이터를 가지고 있으므로, 상기 RLP제어기 131은 상기 2번 블록의 앞 4 바이트로 RLP 프레임을 만든다. 상기 RLP제어기 131은 상기 일련 번호 2번 블록의 일련 번호 0번 데이터 바이트부터 시

작된 데이터 조각으로 RLP 프레임을 만들었으므로 블록 일련 번호 2 그리고 데이터 일련 번호 0을 RLP 프레임 헤더에 표시한다. 상기 RLP제어기 131은 또한 상기 일련 번호 2번 블록의 크기가 6 바이트이므로 블록 끝 지시자를 '0'으로 설정한다.

<150>      상기 다중화/역다중화 제어기 140에서 다시 4 바이트의 데이터를 담을 수 있는 데이터 블록을 요청하면, 상기 RLP제어기 131은 상기 도 7의 RLP 프레임 D와 같이 네번째 RLP 프레임을 생성한다. 상기 도 7에서 일련 번호 2번 블록의 앞 4 바이트 데이터는 이미 모두 전송되었으므로, 상기 RLP제어기 131은 세번째 블록인 일련 번호 2번 블록의 5번째 데이터 바이트부터 RLP 프레임을 생성하는데 사용한다. 상기 도 7에서 일련 번호 2번 블록이 전송하지 않은 2 바이트의 데이터를 가지고 있으므로, 상기 RLP제어기 131은 상기 2번 블록의 데이터만으로는 RLP 프레임을 모두 채울 수 없다. 따라서 상기 RLP제어기 131은 다음 블록인 일련 번호 3번 블록의 데이터를 함께 이용하여 RLP 프레임을 만든다. 즉, 상기 RLP제어기 131은 상기 일련 번호 2번 블록의 뒤 2 바이트와 상기 일련 번호 3번 블록의 앞 2 바이트를 사용하여 총 4 바이트의 데이터를 가진 RLP 프레임을 만든다. 상기 RLP제어기 131은 상기 일련 번호 2번 블록의 일련 번호 4번 데이터 바이트부터 시작된 데이터 조각으로 RLP 프레임을 만들었으므로, 블록 일련 번호 2 그리고 데이터 일련 번호 4를 RLP 프레임 헤더에 표시한다. 상기 RLP제어기 131은 또한 상기 일련 번호 2번 블록의 크기가 6 바이트이고 상기 일련 번호 3번 블록의 끝 부분을 담지 않았으므로 블록 끝 지시자를 '0'으로 설정한다.

<151>      상기 다중화/역다중화 제어기 140에서 다시 4 바이트의 데이터를 담을 수 있는 데이터 블록을 요청하면, 상기 RLP제어기 131은 상기 도 7의 RLP 프레임 E와 같이 다섯번째 RLP 프레임을 생성한다. 상기 도 7에서 일련 번호 3번 블록의 앞 2 바이트 데이터는

이미 모두 전송되었으므로, 상기 RLP제어기 131은 네번째 블록인 일련 번호 3번 블록의 3번째 데이터 바이트부터 RLP 프레임을 생성하는데 사용한다. 상기 도 7에서 일련 번호 3번 블록이 전송하지 않은 3 바이트의 데이터를 가지고 있고 더이상의 데이터가 없으므로, 상기 RLP제어기 131은 상기 3번 블록의 데이터만으로 RLP 프레임을 만든다. 즉, 상기 RLP제어기 131은 상기 일련 번호 3번 블록의 뒤 3 바이트를 사용하여 총 3 바이트의 데이터를 가진 RLP 프레임을 만든다. 상기 RLP제어기 131은 상기 일련 번호 3번 블록의 일련 번호 2번 데이터 바이트부터 시작된 데이터 조각으로 RLP 프레임을 만들었으므로, 블록 일련 번호 3 그리고 데이터 일련 번호 2를 RLP 프레임 헤더에 표시한다. 상기 RLP 제어기 131은 또한 상기 일련 번호 3번 블록의 크기가 6 바이트 보다 작은 4 바이트이고 3번 블록의 마지막 부분을 담았으므로 블록 끝 지시자를 '1'로 설정한다.

<152>        상기 RLP제어기 131은 만일 전달할 데이터가 없으면 빈 데이터 블록을 전달한다.

<153>        상기 RLP제어기 131은 상기와 같은 방법으로 블록을 생성할 때마다 <생성된 블록의 일련 번호, 생성된 블록이 포함하는 데이터 조각의 크기, 생성된 블록이 포함하는 데이터 조각>을 재전송 엔트리(retransmission entry)에 담아서 상기 재전송 버퍼 133에 저장해둔다. 상기 재전송 버퍼 133은 일정 시간 동안 상기 재전송 엔트리를 가지고 있으면서, 상기 RLP제어기 131이 수신측 RLP제어기에게서 재전송 요청을 받았을 때 저장되어 있는 재전송 엔트리 중에서 요청된 블록 일련 번호를 가지는 재전송 엔트리를 상기 RLP 제어기 131에게 전달한다. 상기 RLP제어기 131은 전달된 재전송 엔트리에 저장된 데이터 조각에서 수신측 RLP제어기에서 요청한 데이터 부분을 다시 전송하는 재전송 RLP 프레임을 만든다. 상기 RLP제어기 131은 상기 재전송 RLP 프레임을 만들 때도 새 데이터를 전송하는 RLP 프레임을 만드는 것과 동일한 방법으로 RLP 프레임을 생성한다. 즉, 상기

RLP제어기 131은 수신된 RLP 프레임이 담아온 데이터 조각이 어떤 블록의 몇번째 데이터 부터인지 수신측 RLP제어기가 알 수 있도록 해당하는 데이터 조각의 블록 일련 번호와 데이터 일련 번호를 RLP 프레임에 포함시켜 전송한다. 또한 상기 RLP제어기 131은 새 데이터를 전송할 때와 마찬가지로 상기 RLP 프레임을 만들 때 전송하는 블록의 크기가  $N_{DATA}$  바이트 보다 작은 경우, 수신측 RLP제어기에서 상기 전송된 블록의 끝을 알 수 있도록 해야한다. 또한 상기 RLP제어기 131은 수신된 RLP 프레임이 담아온 데이터 조각의 적어도 첫번째 데이터 바이트는 재전송된 데이터 바이트임을 수신측 RLP제어기가 알 수 있도록 알려야한다. 본 발명에서는 이러한 목적으로 재전송 지시자(retransmit indicator)를 포함시킨다. 즉, 상기 송신측 RLP제어기는 RLP 프레임을 만들 때 RLP에 포함된 첫번째 바이트가 재전송되는 바이트이면, 재전송 지시자를 '1'로 설정하고 그렇지 않을 때는 '0'으로 설정한다. 상기 재전송 지시자가 '1'로 설정된 RLP 프레임을 받으면 수신측 RLP제어기는 재전송된 데이터를 처리하는 절차를 수행하게 된다.

<154>      RLP제어기의 데이터 수신 동작

<155>      상기 RLP제어기 131은 상기 다중화/역다중화 제어기가 수신된 RLP 프레임을 전달할 때마다 수신된 RLP 프레임이 새 데이터를 담고 있는지, 아니면 재전송 데이터를 담고 있는지 살펴본다. 상기한 바와 같이 이러한 작업은 상기 재전송 지시자로 알 수 있다. 만일 수신된 RLP 프레임의 재전송 지시자가 '0'이면 새 데이터를 담은 RLP 프레임이므로, 상기 RLP제어기 131은 상기 수신된 RLP 프레임의 블록 일련 번호와 데이터 일련 번호를

살펴본다. 상기 일련 번호가 현재 수신을 기다리고 있는 블록 일련 번호, 데이터 일련 번호와 같으면 중간에 빠진 새 데이터 바이트는 없는 것이다. 따라서 상기 RLP제어기 131은 수신된 데이터 조각을 상기 수신 데이터 버퍼 124에 바로 저장한다. 만일 같지 않으면, 상기 RLP제어기 131은 송신측 RLP제어기에게 빠진 일련 번호 만큼의 블록과 데이터를 재전송하도록 요청하고, 상기 수신된 새 블록의 데이터 조각은 상기 재정렬 버퍼 138에 저장해둔다.

<156>      상기 RLP제어기 131은 상기 빠진 일련 번호를 다음과 같이 구한다. 상기 2차원 주소 방식에서 상기 RLP제어기 131은 수신을 예상했던 블록 일련 번호와 데이터 일련 번호에서 현재 수신된 새 데이터 조각의 첫번째 바이트의 블록 일련 번호와 데이터 일련 번호 전까지 빠진 일련 번호로 간주한다. 따라서 RLP제어기 131은 상기 수신을 예상했던 블록 일련 번호와 데이터 일련 번호를 상기 NAK 리스트 131에 NAK 엔트리(NAK entry)로 저장하고, 상기 블록 일련 번호와 데이터 일련 번호를 가지는 데이터 바이트의 다음번 데이터 바이트가 가지는 블록 일련 번호와 데이터 일련 번호를 구한다. 상기 RLP제어기 131은 상기 구한 다음번 데이터 바이트가 가지는 블록 일련 번호와 데이터 일련 번호가 상기 수신된 새 데이터 조각의 첫번째 바이트의 블록 일련 번호, 데이터 일련 번호와 같지 않으면 구해진 블록 일련 번호와 데이터 일련 번호를 상기 NAK 리스트 131에 NAK 엔트리(NAK entry)로 저장하고, 다시 다음 데이터 바이트가 가지는 블록 일련 번호와 데이터 일련 번호를 구한 다음 상기 과정을 반복한다. 만일 다음 데이터 바이트가 가지는 블록 일련 번호와 데이터 일련 번호가 상기 수신된 새 데이터 조각의 첫번째 바이트의 블록 일련 번호, 데이터 일련 번호와 같으면 상기 RLP제어기 131은 NAK 엔트리를 추가하는 과정을 마친다.



<157>      상기 2차원 주소 방식에서 블록 일련 번호 B\_SEQ와 데이터 일련 번호 D\_SEQ를 가지는 데이터 바이트가 있을 때 이 데이터 바이트의 다음번 데이터 바이트가 가지는 블록 일련 번호 B\_SEQNEXT와 데이터 일련 번호 D\_SEQNEXT는 다음 <수학식 1>과 같이 구해진다

<158>    **【수학식 1】**

$$D\_SEQNEXT = D\_SEQ + 1 \quad (D\_SEQ \text{ 값이 } (N_{DATA} - 1)$$

와 다른 경우) 또는

$$0 \quad (D\_SEQ \text{ 값이 } (N_{DATA} - 1) \text{와 같은 경우})$$

$$<160> \quad B\_SEQNEXT = B\_SEQ \quad (D\_SEQ \text{ 값이 } (N_{DATA} - 1) \text{와 다른 경우) 또는}$$

$$<161> \quad (B\_SEQ + 1) \text{ modulo } N_{BLOCK} \quad (D\_SEQ \text{ 값이 } (N_{DATA} - 1) \text{와 같은 경우})$$

<162>      상기 <수학식 1>에서 modulo 연산은 modulo 기호의 왼쪽값을 오른쪽값으로 나눈 나머지를 구하는 연산이다.

<163>      상기 RLP제어기 131은 상기 NAK 엔트리를 <빠진 데이터 바이트의 블록 일련 번호, 빠진 데이터 바이트의 데이터 일련 번호, 빠진 데이터 바이트가 수신되었는지를 알려주는 지시자, 재전송 타이머(retransmit timer), 취소 타이머(abort timer)>로 구성하여 상기 NAK 리스트 137에 저장한다.

<164>      상기 RLP제어기 131은 빠진 데이터 바이트들에 대해 NAK 엔트리 추가 작업을 마친 다음, 상기 수신된 새 블록의 데이터 조각의 데이터 바이트 각각에 대해서 상기 <수학식 1>을 사용하여 자신의 블록 일련 번호와 데이터 일련 번호를 부여한다. 즉, 상기 RLP제

여기 131은 상기 수신된 블록 일련 번호와 데이터 일련 번호를 수신된 데이터 조각의 첫 번째 데이터 바이트에게 부여한다. 그런 다음 상기 RLP제어기 131은 수신된 데이터 조각의 각 데이터 바이트들에게 상기 <수학식 1>과 앞 데이터 바이트의 블록 일련 번호, 데이터 일련 번호를 사용하여 자신의 블록 일련 번호와 데이터 일련 번호로 부여한다.

<165>      상기 RLP제어기 131은 수신된 새 블록의 데이터 바이트 마다 일련 번호를 부여한 다음 상기 재정렬 버퍼 138에 수신된 데이터 바이트 각각에 대해서 <수신된 데이터 바이트의 블록 일련 번호, 수신된 데이터 바이트의 데이터 일련 번호, 수신된 데이터 바이트, 수신된 데이터 바이트가 블록의 마지막 데이터 바이트인지를 알려주는 지시자>로 구성된 재정렬 버퍼 엔트리(resequencing buffer entry)를 저장하여, 빠진 부분이 도착하면 이미 수신된 데이터들과 함께 전체 데이터를 순서대로 정렬할 수 있도록 한다. 특히 송신측 RLP제어기에서 수신된 블록의 크기가  $N_{DATA}$  바이트 보다 작으며 이번에 수신된 RLP 프레임이 상기 블록의 마지막 끝 부분을 담고 있다는 것을 알려주면, 상기 RLP제어기 131은 수신된 마지막 데이터 바이트에 대한 상기 재정렬 버퍼 엔트리를 만들 때 수신된 데이터 바이트가 블록의 마지막 데이터 바이트인지를 알려주는 지시자에 마지막임을 표시한다.

<166>      상기 새 데이터를 담은 RLP 프레임을 처리한 다음, 상기 RLP제어기 131은 다음과 같이 다음번 새 데이터의 블록 일련 번호와 데이터 일련 번호를 계산한다. 즉, 만일 송신측 RLP제어기에서 수신된 블록의 크기가  $N_{DATA}$  바이트 보다 작으며 이번에 수신된 RLP 프레임이 상기 블록의 마지막 끝 부분을 담고 있다는 것을 알려주면, 상기 RLP제어기 131은 블록 일련 번호를 다음 새 블록의 일련 번호로 설정하고 데이터 일련 번호를 0으로 설정한다. 상기 다음 새 블록의 일련 번호는 상기한 바와 같이 앞의 블록의 일련 번

호에서 하나 증가된 일련 번호가 된다. 만일 앞의 블록이 ( $N_{BLOCK} - 1$ )의 일련 번호를 받았다면, 새 블록의 일련 번호는 일련 번호 0번이 된다. 만일 송신측 RLP제어기에서 수신된 블록의 크기가  $N_{DATA}$  바이트와 같거나 또는 이번에 수신된 RLP 프레임이 크기가  $N_{DATA}$  바이트 보다 작은 블록의 마지막 끝 부분을 담고 있지 않다는 것을 알려주면, 상기 RLP 제어기 131은 블록 일련 번호를 그대로 두고 데이터 일련 번호만 수신된 데이터 조각의 크기만큼 증가시킨다. 만일 증가된 데이터 일련 번호가  $N_{DATA}$  보다 커지게 되면 이번 블록의 데이터는 모두 수신된 것이므로 상기 RLP제어기 131은 블록 일련 번호를 다음 새 블록의 일련 번호로 설정하고 데이터 일련 번호를 상기 증가된 데이터 일련 번호와  $N_{DATA}$  의 차로 설정한다.

<167>      상기 RLP제어기 131은 수신된 RLP 프레임의 재전송 지시자가 '1'이면 수신된 RLP 프레임이 재전송된 데이터 조각을 가지고 있으므로 상기 수신된 블록의 데이터 조각의 데이터 바이트 각각에 대해서 상기 <수학식 1>을 사용하여 자신의 블록 일련 번호와 데이터 일련 번호를 부여한다. 즉, 상기 RLP제어기 131은 상기 수신된 블록 일련 번호와 데이터 일련 번호를 수신된 데이터 조각의 첫번째 데이터 바이트에게 부여한다. 그런 다음 상기 RLP제어기 131은 수신된 데이터 조각의 각 데이터 바이트들에게 상기 <수학식 1>과 앞 데이터 바이트의 블록 일련 번호, 데이터 일련 번호를 사용하여 자신의 블록 일련 번호와 데이터 일련 번호로 부여한다.

<168>      상기 과정이 끝나면 상기 RLP제어기 131은 각 수신된 데이터 바이트에 대해서 가지고 있는 블록 일련 번호와 데이터 일련 번호가 일치하는 NAK 리스트 137의 엔트리를 찾는다. 상기 찾아진 NAK 엔트리가 이미 수신된 것으로 되어 있으면, 상기 RLP제어기 131은 상기 데이터 바이트를 버린다. 만일 상기 찾아진 NAK 엔트리가 아직 수신되지 않은

엔트리 중에서 가장 오래된 것이라면 상기 수신된 데이터 바이트까지는 모두 수신된 것이므로, 상기 RLP제어기 131은 상기 재정렬 버퍼 138에 저장된 수신된 연속적인 데이터 바이트들을 일련 번호 순서대로 정렬한 다음 상기 수신 데이터 버퍼 124로 보낸다. 상기 NAK 엔트리가 아직 수신되지 않은 엔트리 중에서 가장 오래된 것이 아니라면 상기 RLP제어기 131은 일단 NAK 엔트리 <빠진 데이터 바이트의 블록 일련 번호, 빠진 데이터 바이트의 데이터 일련 번호, 빠진 데이터 바이트가 수신되었는지를 알려주는 지시자, 재전송 타이머(retransmit timer), 취소 타이머(abort timer)>의 빠진 데이터 바이트가 수신되었는지를 알려주는 지시자를 수신되었음으로 설정하고, 수신된 데이터 바이트를 상기 재정렬 버퍼 138에 저장한다. 상기 RLP제어기 131은 상기 데이터 바이트에 대해서 새로운 재정렬 버퍼 엔트리 <수신된 데이터 바이트의 블록 일련 번호, 수신된 데이터 바이트의 데이터 일련 번호, 수신된 데이터 바이트, 수신된 데이터 바이트가 블록의 마지막 데이터 바이트인지를 알려주는 지시자>를 만들어서 상기 재정렬 버퍼 138에 저장해둔다. 상기 저장된 데이터 부분은 빠진 앞부분의 데이터 부분이 도착하여 연속적으로 일련 번호를 가지는 프레임들이 수신되면 상기한 바와 같이 수신 데이터 버퍼 124로 보내어지게 된다.

<169>      상기 RLP제어기 131은 수신된 RLP 프레임에 대한 처리가 끝나면 NAK 리스트 137의 각 NAK 엔트리에 대해서 재전송 요청에 대한 타이머를 조절하는 작업을 수행한다. 즉, 수신된 RLP 프레임 중에서 새로 전송된 데이터 바이트가 있거나 또는 송신측 RLP제어기가 아무것도 전송하지 않은 것으로 판단되면, 송신측 RLP제어기는 아무런 재전송 요청을 가지고 있지 않다는 것을 의미하므로, 상기 RLP제어기 131은 재전송 요청을 다시 하여야만 한다. 따라서 상기 RLP제어기 131은 각 NAK 엔트리에서 아직 수신되지 않은 NAK 엔

트리의 재전송 타이머 또는 취소 타이머값을 하나 감소시키고 이 값이 0이 되면 다시 재전송 요청을 하거나 아니면 재전송 요청을 포기하고 수신된 나머지 부분만을 상기 수신 데이터 버퍼 124로 전달한다.

<170>      RLP제어기의 데이터 송신전 동작

<171>      RLP제어기 131은 동작을 시작하기 전에 상기 도 3과 도 4에서 보여주는 것과 같이 L\_V(S) 레지스터 132와, L\_V(R) 레지스터 136, L\_V(N) 레지스터 135, E 레지스터 134를 '0'으로 설정한다. RLP제어기 131은 동작을 시작하기 전에 재전송 버퍼 133과, NAK 리스트 137, 그리고 재정렬 버퍼 138을 모두 비운다. RLP제어기 131은 마지막으로 재전송에 관련된 모든 타이머를 끈다.

<172>      상기 RLP제어기 131이 상기 다중화/역다중화 제어기에게 전달할 수 있는 데이터 블록의 종류는 도 8, 도 9와 같다.

<173>      상기 도 8은 기본 채널로 전송할 수 있는 RLP 프레임을 보여주고, 상기 도 9는 부가 채널로 전송할 수 있는 RLP 프레임을 보여준다. 본 발명에서 RLP 프레임은 다음과 같이 부르겠다. 상기 도 8의 SYNC, SYNC/ACK, ACK, 또는 하기 <표 14>의 NAK 프레임을 '제어 프레임(control frame)'이라고 하고, 데이터를 담은 프레임을 '데이터 프레임(data frame)'이라고 칭하겠다. 상기 데이터 프레임은 새로 전송되는 데이터 부분을 담고 있는 '새 데이터 프레임(new data frame)'과, 재전송되는 데이터 부분을 담고 있는 '재전송 데이터 프레임(retransmitted data frame)'으로 나누어진다. 상기 12 비트 SEQ 필드만

가지는 프레임은 아이들 프레임이라고 하여 상기 제어 프레임, 데이터 프레임과는 별도로 칭하겠다.

<174>      상기 도 8에서 기본 채널로 전송되는 데이터 블록은 제어 프레임, 데이터 프레임, 또는 아이들 프레임이 가능하다.

<175>      상기 도 9에서 부가 채널로 전송되는 데이터 블록은 데이터 프레임만 가능하다.

<176>      상기 RLP제어기 131은 데이터를 전송하기 전에 재설정 과정을 수행한다. 상기 RLP 제어기 131은 SYNC 프레임을 데이터 블록으로 다중화/역다중화 제어기 140에게 계속 전달한다.

<177>      상기 RLP제어기 131은 SYNC 프레임을 상기 다중화/역다중화 제어기 140에게서 받으면, SYNC/ACK 프레임을 상기 다중화/역다중화 제어기 140에게 SYNC/ACK 프레임을 물리 채널 프레임이 수신되었고 빈 데이터 블록이 아니며 SYNC 프레임이 아닐 때까지 계속 전달한다.

<178>      상기 RLP제어기 131은 SYNC/ACK 프레임을 받으면, 상기 다중화/역다중화 제어기 140에게 ACK 프레임을 전달한다. 상기 RLP제어기 131은 ACK 프레임을 상기 다중화/역다중화 제어기 140에서 물리 채널 프레임이 수신되었고 빈 데이터 블록이 아니며 SYNC/ACK 프레임이 아닐 때까지 계속 전달한다. 상기 RLP제어기 131은 물리 채널 프레임이 수신되었고 수신된 데이터 블록이 빈 데이터 블록이 아니며 SYNC/ACK 프레임이 아닌 RLP 프레임을 가지고 있으면 데이터 송신을 시작한다.

<179>      상기 RLP제어기 131은 ACK 프레임을 받으면 데이터 송신을 시작한다. 상기 RLP제어기 131은 상기 다중화/역다중화 제어기 140에게 SYNC, SYNC/ACK, ACK 프레임을 제외한

다른 프레임을 전달할 수 있다.

<180>      RLP제어기의 기본 채널 데이터 송신 동작

<181>      상기 송신측 다중화/역다중화 제어기는 상기 RLP제어기 131에게 기본 채널로 전송할 데이터 블록을 요청하여 상기 RLP제어기 131이 프레임을 생성하도록 한다. 상기 송신측 다중화/역다중화 제어기는 상기 RLP제어기에게 데이터 블록을 요청하면서 상기 RLP제어기 131이 만들어내야하는 데이터 블록의 크기 T를 함께 알려준다. 본 발명에서는 상기 송신측 다중화/역다중화 제어기 140에서 요청한 데이터 블록의 크기 T가 언제나 전송을 집합 1에서 171 비트 또는 전송을 집합 2에서 266 비트라고 가정한다.

<182>      상기 RLP제어기 131은 기본 채널로 전송할 데이터 블록을 만들 때 다음의 우선 순위에 따라 데이터 블록을 만든다.

<183>      첫째, 제어 프레임 (SYNC, SYNC/ACK, ACK, NAK frame)

<184>      둘째, 재전송 데이터 프레임

<185>      셋째, 새 데이터 프레임

<186>      넷째, 아이들 프레임

<187>      상기 RLP제어기 131은 전송할 제어 프레임이 있으면 다음과 같이 제어 프레임을 만든다. SYNC, SYNC/ACK, ACK 프레임의 경우, 상기 도 8의 (a)에 도시된 바와 같이 SEQ 필드를 '00000000'으로, CTL 필드를 프레임 종류에 따라서 설정한 후 FCS 필드를 붙인다.

FCS 필드는 RFC-1662에 의하여 규정된 다항식(polynomial)에 의하여 만들어지는 16 비트 프레임 점검 비트열(frame check sequence)이다. 상기 FCS 필드는 앞에 오는 모든 비트에 대하여 만들어진다. 상기 RLP제어기 131은 상기 다중화/역다중화 제어기에서 알려준 크기에 맞추어 상기 FCS 필드 다음을 모두 '0'으로 설정한다. 상기 RLP제어기 131은 데이터 블록의 마지막 1 비트인 TYPE 필드를 '1'로 설정한 다음, 생성된 데이터 블록을 상기 다중화/역다중화 제어기에게 전달한다.

<188>      상기 RLP제어기 131은 다시 전송되어야하는 데이터들이 있으면 NAK 프레임을 생성하여 상기 다중화/역다중화 제어기에게 전달한다. 상기 NAK 프레임은 하기 <표 14>와 같은 구조를 가진다.

<189>      【표 14】

NAK 프레임

필드	길이
SEQ	8 비트
CTL	8 비트
NAK_COUNT	2 비트
NAK_COUNT + 1만큼 다음의 필드들이 들어간다:	
NAK_TYPE_AND_UNIT	4 비트
NAK_TYPE_AND_UNIT이 '0001'이면 다음의 필드들이 들어간다:	
FIRST	12 비트
LAST	12 비트
NAK_TYPE_AND_UNIT이 '0010' 또는 <표 15>나 <표 16>에 정의된 값이면 다음의 필드들이 들어간다.	
NAK_MAP_SEQ	12 비트
NAK_MAP	8 비트
NAK_TYPE_AND_UNIT이 '1111'이면 다음의 필드들이 들어간다.	
NAK_MAP_SEQ	12 비트
FIRST	13 비트
LAST	13 비트
어떠한 NAK_TYPE에 대해서도 다음의 필드들이 들어간다:	
PADDING_1	가변길이
FCS	16 비트
PADDING_2	가변길이
TYPE	1 비트



- <190>      상기 <표 14>에서 상기 RLP제어기 131은 NAK 프레임을 다음과 같이 만든다. 상기 <표 14>의 SEQ 필드는 '00000000'으로, CTL 필드는 '00011100'으로, 그리고 TYPE 필드는 '1'로 설정한다.
- <191>      상기 RLP제어기 131은 상기 NAK 프레임에 포함될 재전송 요구의 수에서 하나를 뺀 값을 상기 NAK\_COUNT 필드값으로 한다.
- <192>      상기 RLP제어기 131은 상기 NAK\_COUNT 필드값에 하나를 더한 만큼 재전송 요구를 넣는다. 상기 재전송 요구는 상기 <표 14>에서 알 수 있듯이 NAK\_TYPE\_AND\_UNIT 필드와 이 필드 값에 따라서 FIRST, LAST 필드 또는 NAK\_MAP\_SEQ, NAK\_MAP 필드로 구성된다. 상기 재전송 요구의 NAK\_TYPE\_AND\_UNIT 필드를 '0001'으로 설정한 경우, 상기 RLP제어기 131은 연속적으로 재전송을 요청하는 첫번째 블록의 블록 일련 번호를 FIRST 필드에 넣고 마지막 블록의 블록 일련 번호를 LAST 필드에 넣는다.
- <193>      상기 RLP제어기 131은 상기 NAK\_TYPE\_AND\_UNIT을 '0010'과 같이 설정할 수 있다. 상기 RLP제어기 131이 상기 NAK\_TYPE\_AND\_UNIT 필드를 '0010'과 같이 설정하면 재전송 요구는 NAK\_MAP\_SEQ 필드와 NAK\_MAP 필드를 가지는 블록에 대한 NAK MAP 방식이 된다. 상기 RLP제어기 131은 여러 개의 불연속적인 블록들에 대한 재전송을 요청할 때 상기 NAK\_TYPE\_AND\_UNIT을 '0010'과 같이 설정하고 다음과 같이 NAK MAP을 채운다. 상기 NAK MAP은 재전송을 요청할 첫번째 블록의 블록 일련 번호를 NAK\_MAP\_SEQ 필드에 넣고,  $(\text{NAK\_MAP\_SEQ} + n) \bmod 2^{12}$ 의 일련 번호에 해당하는 블록에 대해서 재전송을 요청해야 하면 NAK\_MAP의 최상위에서 n번째 비트를 '1'로 설정한다. 상기 n 값은 1에서 8까지의 값을 가질 수 있다. 예를 들어, NAK\_TYPE\_AND\_UNIT 필드가 '0010'이고 전송을 집합이

1일 때 NAK\_MAP\_SEQ가 0으로 설정되어 있고 NAK\_MAP이 '10000000'이면 이를 수신한 RLP 제어기는 블록 일련 번호 0인 블록과 1인 블록을 재전송하여야 한다.

<194>      상기 RLP제어기 131은 블록의 일부분에 대한 재전송을 요청할 때 상기 NAK\_TYPE\_AND\_UNIT을 다음 <표 15> 또는 <표 16>과 같이 설정할 수 있다. 상기 RLP제어기 131이 상기 NAK\_TYPE\_AND\_UNIT 필드를 <표 15> 또는 <표 16>과 같이 설정하면 재전송 요구는 NAK\_MAP\_SEQ 필드와 NAK\_MAP 필드를 가지는 블록의 일부분에 대한 NAK MAP 방식이 된다.

<195>    【표 15】

NAK\_TYPE\_AND\_UNIT 필드 (전송율 집합 1)

필드 값	일련 번호 수
'0011'	19개
'0100'	41개
'0101'	42개
'0110'	90개
'0111'	186개
'1000'	378개
'1001'	762개
'1010'	1530개

<196>

【표 16】

NAK\_TYPE\_AND\_UNIT 필드 (전송을 집합 2)

필드 값	일련 번호 수
'0011'	31개
'0100'	65개
'0101'	66개
'0110'	138개
'0111'	282개
'1000'	318개
'1001'	570개
'1010'	1146개
'1011'	2586개

<197>        상기 RLP제어기 131은 상기 <표 15> 또는 <표 16>을 바탕으로 NAK MAP을 채운다.

상기 NAK MAP은 재전송을 요청할 블록의 블록 일련 번호를 NAK\_MAP\_SEQ 필드에 넣고 상기 <표 15> 또는 <표 16>에 표시된 단위로 재전송을 요청하는 블록의 데이터 조각의 데이터 바이트 일련 번호들을 NAK\_MAP에 표시한다. 상기 RLP제어기 131은 상기 NAK MAP을 사용하여, NAK\_TYPE\_AND\_UNIT 필드에 의해서 결정된 단위를 U일 때, 상기 NAK\_MAP의 최상위에서 n번째 비트가 '1'일 때마다  $((n - 1) * U)$ 에서  $(n * U - 1)$ 까지의 데이터 일련 번호에 해당하는 상기 블록에 속하는 데이터 바이트에 대해서 재전송을 요청한다. 상기 n 값은 1에서 8까지의 값을 가질 수 있다. 상기 NAK MAP에서 예를 들어

NAK\_TYPE\_AND\_UNIT 필드가 '0011'이고 전송을 집합이 1일 때 NAK\_MAP\_SEQ가 0으로 설정되어 있고 NAK\_MAP이 '10000000'이면 이를 수신한 RLP제어기는 블록 일련 번호 0번인 블록의 데이터 부분에서 데이터 일련 번호 0에서 18까지의 데이터 바이트, 즉 첫번째에서 19번째까지의 데이터 바이트를 재전송하여야 한다.

<198>        상기 RLP제어기 131은 블록에 속하는 데이터 바이트들에 대한 재전송을 요청할 때 상기 NAK\_TYPE\_AND\_UNIT을 '1111'으로 설정할 수 있다. 상기 RLP제어기 131이 상기

NAK\_TYPE\_AND\_UNIT 필드를 '1111'로 설정하면 재전송 요구는 NAK\_MAP\_SEQ 필드와 FIRST, LAST 필드를 가진다. 상기 RLP제어기 131은 재전송을 요청할 데이터 조각을 담았었던 블록의 블록 일련 번호를 NAK\_MAP\_SEQ 필드에 넣고 재전송을 요청할 데이터 조각의 첫번째 바이트의 데이터 일련 번호를 FIRST 필드에, 그리고 마지막 바이트의 데이터 일련 번호를 LAST 필드에 넣는다.

<199>        상기 RLP제어기 131은 상기 재전송 요구를 상기 NAK\_COUNT 필드값에 하나를 더한 만큼 만들어 넣은 다음 FCS 필드를 바이트 정렬로 맞추기 위하여 '0'으로 패딩을 하고 FCS 필드를 채운다. 상기 FCS 필드는 RFC-1662에 의하여 규정된 다항식(polynomial)에 의하여 만들어지는 16 비트 프레임 점검 비트열(frame check sequence)이다. 상기 FCS 필드는 앞에 오는 모든 비트에 대하여 만들어진다. 상기 FCS 필드를 넣은 다음 상기 RLP 제어기 131은 데이터 블록의 나머지 부분에 대해서 '0'을 채운다.

<200>        RLP제어기 131이 기본 채널로 전송할 새 데이터 프레임을 생성할 때는 다음과 같이 동작한다.

<201>        상기 RLP제어기 131은 새 데이터 프레임을 가장 처음 생성할 때 또는 이전에 정의 하였던 블록이 가지고 있는 모든 데이터 바이트를 전송하였으면 데이터열의 남은 데이터에 대해서 새 블록을 정의한다. 본 발명에서는 새 블록을 정의할 때 상기 다중화/역다중화 제어기 140이 요청한 데이터 블록에 담을 수 있는 데이터 크기만큼 블록이 데이터 조각을 가지도록 정의하는 방식을 모드 A(mode A)라고 부르고, 반면 상기 다중화/역다중화 제어기 140이 요청한 데이터 블록의 크기에 관계없이 언제나 N<sub>DATA</sub> 바이트의 데이터를 가지도록 정의하는 방식을 모드 B(mode B)라고 부르겠다. 상기 RLP제어기 131은 새 블록을 정의할 때마다 모드 A와 모드 B 중에서 가장 적절한 방식을 선택할 수 있다. 또한 본

발명에서는 상기  $N_{BLOCK}$ 과  $N_{DATA}$ 를 각각  $2^8$ ,  $2^9$ 으로 정의한다.

<202> 도 8에서는 모드 A와 B에서 기본 채널로 전송할 수 있는 사용가능한 프레임 포맷들을 보여주고 있다. 상기 모드 A에서 생성된 새 데이터 프레임이 가질 수 있는 포맷이 상기 도 8의 (b)와 (c) 포맷이다. 상기 모드 B에서 생성된 새 데이터 프레임이 가질 수 있는 포맷은 상기 도 8의 (d)와 (e) 포맷이다.

<203> 상기 RLP제어기 131은 새 데이터를 담은 기본 채널 RLP 프레임을 생성할 때 21 비트 일련 번호 레지스터 L\_V(S) 132를 사용한다. 상기 RLP제어기 131은 상기 일련 번호 레지스터 L\_V(S) 132의 21 비트 값에서 상위 12 비트를 블록 일련 번호 B\_SEQ로, 그리고 하위 9 비트를 데이터 일련 번호 D\_SEQ로 결정한다.

<204> 상기 RLP제어기 131은 상기 과정에서 구해진 D\_SEQ 값이 0이 아니면 아직 전송하지 않은 블록의 데이터 부분이 있는 것으로 판단한다. 즉, 상기 블록 일련 번호 B\_SEQ에 해당하는 블록의 데이터 일련 번호 D\_SEQ에서부터 ( $N_{DATA} - 1$ )까지의 데이터 바이트들이 전송되어야 한다고 판단한다. 따라서 상기 RLP제어기 131은 상기 데이터 조각을 전송하기 위해서 모드 B에서 사용할 수 있는 프레임 포맷인 도 8의 (d)와 (e) 포맷을 사용한다. 상기 도 8의 (d) 포맷은 다음의 조건을 만족할 때 사용할 수 있다.

<205> 첫째, 아직 전송되지 않은 데이터열의 데이터가 상기 도 8의 (d) 포맷으로 보낼 수 있는 데이터 개수 B, 즉 전송을 집합 1에서 19 바이트 또는 전송을 집합 2에서 31 바이트 이상이며,

<206> 둘째, 상기 과정에서 구해진 D\_SEQ 값이 상기 도 8의 (d) 포맷으로 보낼 수 있는 데이터 개수 B, 즉 전송을 집합 1에서 19 또는 전송을 집합 2에서 31의 배수일 때 상기

도 8의 (d) 포맷을 사용한다.

<207>      상기 RLP제어기 131은 상기 도 8의 (d) 포맷을 사용하여 새 데이터를 전송할 때 먼저 SEQ 필드를 상기 데이터 부분의 블록 일련 번호인 B\_SEQ 값의 하위 8 비트로 설정한다. 상기 RLP제어기 131은 또한 상기 D\_SEQ 값을 상기 도 8의 (d) 포맷으로 보내는 데이터 개수 B로 나눈 몫으로 SEG 필드를 설정한다.

<208>      상기 RLP제어기 131은 상기 도 8의 (d) 포맷을 사용하여 새 데이터를 전송할 때 TYPE 필드를 상기 도 8과 같이 '1'로, 재전송 지시자인 REXMIT 필드를 '0'으로, CTL 필드를 '01'로 설정한 다음, DATA 필드를 전송할 데이터로 채운다.

<209>      상기 RLP제어기 131은 상기 도 8의 (e) 포맷을 사용하여 새 데이터를 전송할 때 먼저 SEQ 필드를 상기 데이터 부분의 블록 일련 번호인 B\_SEQ 값의 하위 8 비트로 설정한다. 상기 RLP제어기 131은 또한 상기 D\_SEQ 값으로 SEG 필드를 설정한다.

<210>      상기 RLP제어기 131은 상기 도 8의 (e) 포맷을 사용하여 새 데이터를 전송할 때 TYPE 필드를 상기 도 8과 같이 '1'로, 재전송 지시자인 REXMIT 필드를 '0'으로, CTL 필드를 '1'로 설정하고, DATA 필드를 전송할 데이터로 채운 다음, LEN 필드를 채워진 데이터의 바이트 수로 설정한다.

<211>      상기 RLP제어기 131은 상기 도 8의 (d) 또는 (e) 포맷을 사용하여 새 데이터를 전송할 때 블록 끝 지시자인 END 필드를 다음 조건이 만족되면 '1'로 그렇지 않으면 '0'으로 설정한다.

- <212> 첫째, 전송을 마친 후 더 이상 전송하지 않은 데이터가 데이터열에 남아 있지 않고.
- <213> 둘째, 상기 D\_SEQ에 전송한 데이터 개수 N을 더한 값이  $N_{DATA}$  또는 그 배수 중 하나와 같지 않을 때 블록 끝 지시자인 END 필드를 '1'로 설정한다.
- <214> 상기 RLP제어기 131은 상기 과정에서 새 데이터 프레임을 전송하면 다음과 같이 L\_V(S) 레지스터 132를 수정한다. 상기 RLP제어기 131은 상기 과정에서 전송된 새 데이터의 크기가 N이면 상기 L\_V(S) 레지스터 132에 N 값을 더한다.
- <215> 상기 RLP제어기 131은 상기 D\_SEQ 값이 0이면 새 블록을 정의한다. 상기한 바와 같이 이 작업은 모드 A일 때와 모드 B일 때 수행하는 절차가 달라지게 된다. 만일 상기 RLP제어기 131이 모드 A에 따라서 새 블록을 정의하게 되면 상기 RLP제어기 131은 다음과 같이 동작한다.
- <216> 상기 RLP제어기 131은 모드 A에서 새 블록의 크기를 상기 도 8의 (b)로 보낼 수 있는 데이터 개수 B, 즉 전송을 집합 1에서 20 바이트 또는 전송을 집합 2에서 32 바이트와 아직 전송되지 않은 데이터열의 데이터 수 중에서 최소값 M으로 정한다. 상기 RLP제어기 131은 상기 구해진 최소값 M이 상기 도 8의 (b)로 보낼 수 있는 데이터 개수 B와 같으면 상기 도 8의 (b) 포맷을 사용하고 그렇지 않으면 상기 도 8의 (c) 포맷을 사용한다.
- <217> 상기 RLP제어기 131은 상기 도 8의 (b) 포맷을 사용하여 새 데이터를 전송할 때 먼저 SEQ 필드를 상기 데이터 부분의 블록 일련 번호인 B\_SEQ 값의 하위 8 비트로 설정한

다. 상기 도 8의 (b) 포맷으로 전달되는 데이터는 언제나 데이터 일련 번호가 0부터 시작하고 그 크기가  $N_{DATA}$  보다 작으므로 블록 끝 지시자도 언제나 '1'인 것에 주의하여야 한다. 따라서 상기 도 8의 (b) 포맷에서는 상기 데이터 일련 번호와 블록 끝 지시자를 포함하지 않았지만 수신측 RLP제어기는 상기 도 8의 (b) 포맷을 수신하면 데이터 일련 번호를 0으로, 블록 끝 지시자를 '1'로 인식한다.

<218>      상기 RLP제어기 131은 상기 도 8의 (b) 포맷을 사용하여 새 데이터를 전송할 때 TYPE 필드를 상기 도 8과 같이 '0'로, 재전송 지시자인 REXMIT 필드를 '0'으로 설정한 다음, DATA 필드를 전송할 데이터로 채운다.

<219>      상기 RLP제어기 131은 상기 도 8의 (c) 포맷을 사용하여 새 데이터를 전송할 때 먼저 SEQ 필드를 상기 데이터 부분의 블록 일련 번호인 B\_SEQ 값의 하위 8 비트로 설정한다. 상기 도 8의 (c) 포맷으로 전달되는 데이터는 언제나 데이터 일련 번호가 0부터 시작하고 그 크기가  $N_{DATA}$  보다 작으므로 블록 끝 지시자도 언제나 '1'인 것에 주의하여야 한다. 따라서 상기 도 8의 (c) 포맷에서는 상기 데이터 일련 번호와 블록 끝 지시자를 포함하지 않았지만 수신측 RLP제어기는 상기 도 8의 (c) 포맷을 수신하면 데이터 일련 번호를 0으로, 블록 끝 지시자를 '1'로 인식한다.

<220>      상기 RLP제어기 131은 상기 도 8의 (c) 포맷을 사용하여 새 데이터를 전송할 때 TYPE 필드를 상기 도 8과 같이 '1'로, 재전송 지시자인 REXMIT 필드를 '0'으로, CTL 필드를 '001'로 설정한 다음, DATA 필드를 전송할 데이터로 채우고 LEN 필드를 채워진 데이터의 바이트 수로 설정한다.

<221>      상기 RLP제어기 131은 상기 과정에서 새 데이터 프레임을 전송하면 다음과



같이 L\_V(S) 레지스터 132를 수정한다. 상기 RLP제어기 131은 상기 L\_V(S) 레지스터 132에 N<sub>DATA</sub> 값을 더한다.

<222> 만일 RLP제어기 131이 모드 B에 따라서 새 블록을 정의하게 되면 상기 RLP제어기 131은 다음과 같이 동작한다. 상기 RLP제어기 131은 모드 B에서 새 블록의 크기를 N<sub>DATA</sub> 바이트와 아직 전송되지 않은 데이터열의 데이터 수 중에서 최소값 M으로 정한다. 상기 RLP제어기 131은 새 블록의 크기를 정하면 아직 전송되지 않은 데이터열의 데이터에서 정해진 블록 크기 만큼의 데이터를 새 블록이 가지고 있는 것으로 정의한다. 상기 RLP제어기 131은 상기와 같이 새 블록을 정의하면 정의된 블록에 대해서 상기 블록의 남은 부분을 전송하는 절차를 수행한다.

<223> 상기 RLP제어기 131은 새로 정의한 블록들을 수신측의 재전송 요청에 대비하여 재전송 버퍼 133에 저장해둔다. 상기 RLP제어기 131은 <생성된 블록의 일련 번호, 생성된 블록이 포함하는 데이터 조각의 크기, 생성된 블록이 포함하는 데이터 조각>을 재전송 엔트리(retransmission entry)에 담아서 상기 재전송 버퍼 133에 저장한다. 상기 RLP제어기 131은 수신측에서 재전송을 요청한 경우, 요청된 블록 일련 번호 및 데이터 일련 번호에 해당하는 데이터 바이트를 상기 재전송 버퍼 133에서 찾아서 재전송하여야 한다. 상기 RLP제어기 131은 재전송 요청된 블록 일련 번호를 블록 일련 번호 B\_SEQ로, 그리고 전송할 데이터 조각의 첫번째 바이트의 데이터 일련 번호로 데이터 일련 번호 D\_SEQ로 결정한다.

<224> 상기 RLP제어기 131은 재전송 데이터 프레임을 보낼 때는 상기 도 8의 (b), (c), (d), (e) 포맷을 사용할 수 있다. 상기 RLP제어기 131은 상기 도 8의 (b) 또는 (c) 포맷

을 다음 조건이 만족될 때 사용한다.

- <225> 첫째, 재전송 요청이 블록 전체에 대한 것이고,
- <226> 둘째, 재전송될 블록의 크기가 상기 도 8의 (b) 포맷으로 보낼 수 있는 데이터 개수 B, 즉 전송을 집합 1에서 20 바이트 또는 전송을 집합 2에서 32 바이트와 같거나 작은 경우 상기 RLP제어기 131은 상기 도 8의 (b) 또는 (c) 포맷을 사용한다.
- <227> 상기 RLP제어기 131은 상기 도 8의 (b) 또는 (c) 포맷을 사용하여 상기 블록을 재전송할 때 블록의 크기가 상기 도 8의 (b)로 보낼 수 있는 데이터 개수 B와 같으면 상기 도 8의 (b) 포맷을 사용하고 그렇지 않으면 상기 도 8의 (c) 포맷을 사용한다.
- <228> 상기 RLP제어기 131은 상기 도 8의 (b) 포맷을 사용하여 상기 블록을 재전송할 때 먼저 SEQ 필드를 상기 데이터 부분의 블록 일련 번호인 B\_SEQ 값의 하위 8 비트로 설정한다.
- <229> 상기 RLP제어기 131은 상기 도 8의 (b) 포맷을 사용하여 상기 블록을 재전송할 때 TYPE 필드를 상기 도 8과 같이 '0'로, 재전송 지시자인 REXMIT 필드를 '1'로 설정한 다음, DATA 필드를 재전송할 데이터로 채운다.
- <230> 상기 RLP제어기 131은 상기 도 8의 (c) 포맷을 사용하여 상기 블록을 재전송할 때 먼저 SEQ 필드를 상기 데이터 부분의 블록 일련 번호인 B\_SEQ 값의 하위 8 비트로 설정한다.
- <231> 상기 RLP제어기 131은 상기 도 8의 (c) 포맷을 사용하여 상기 블록을 재전송할 때 TYPE 필드를 상기 도 8과 같이 '1'로, 재전송 지시자인 REXMIT 필드를 '1'으로, CTL 필

드를 '001'로 설정한 다음, DATA 필드를 전송할 데이터로 채우고 LEN 필드를 채워진 데이터의 바이트 수로 설정한다.

<232>      상기 RLP제어기 131은 상기 조건을 만족시키지 못하면 상기 도 8의 (d) 또는 (e) 포맷을 사용한다. 상기 RLP제어기 131은 상기 도 8의 (d) 포맷은 다음의 조건을 만족할 때 사용할 수 있다.

<233>      첫째, 재전송할 블록의 데이터 조각의 크기가 상기 도 8의 (d) 포맷으로 보낼 수 있는 데이터 개수 B, 즉 전송을 집합 1에서 19 바이트 또는 전송을 집합 2에서 31 바이트 이상이며,

<234>      둘째, 재전송할 블록의 데이터 조각의 첫번째 데이터 바이트의 데이터 일련 번호 D\_SEQ 값이 상기 도 8의 (d) 포맷으로 보낼 수 있는 데이터 개수 B, 즉 전송을 집합 1에서 19 또는 전송을 집합 2에서 31의 배수일 때 상기 도 8의 (d) 포맷을 사용한다.

<235>      상기 RLP제어기 131은 상기 도 8의 (d) 포맷을 사용하여 상기 블록의 데이터 조각을 재전송할 때 먼저 SEQ 필드를 상기 데이터 부분의 블록 일련 번호인 B\_SEQ 값의 하위 8 비트로 설정한다. 상기 RLP제어기 131은 또한 상기 D\_SEQ 값을 상기 도 8의 (d) 포맷으로 보내는 데이터 개수 B로 나눈 몫으로 SEG 필드를 설정한다.

<236>      상기 RLP제어기 131은 상기 도 8의 (d) 포맷을 사용하여 상기 블록의 데이터 조각을 재전송할 때 TYPE 필드를 상기 도 8과 같이 '1'로, 재전송 지시자인 REXMIT 필드를 '1'으로, CTL 필드를 '01'로 설정한 다음, DATA 필드를 전송할 데이터로 채운다.

<237>      상기 RLP제어기 131은 상기 도 8의 (e) 포맷을 사용하여 상기 블록의 데이터 조각

을 재전송할 때 먼저 SEQ 필드를 상기 데이터 부분의 블록 일련 번호인 B\_SEQ 값의 하위 8 비트로 설정한다. 상기 RLP제어기 131은 또한 상기 D\_SEQ 값으로 SEG 필드를 설정한다.

<238>      상기 RLP제어기 131은 상기 도 8의 (e) 포맷을 사용하여 상기 블록의 데이터 조각을 재전송할 때 TYPE 필드를 상기 도 8과 같이 '1'로, 재전송 지시자인 REXMIT 필드를 '1'으로, CTL 필드를 '1'로 설정하고, DATA 필드를 전송할 데이터로 채운 다음, LEN 필드를 채워진 데이터의 바이트 수로 설정한다.

<239>      상기 RLP제어기 131은 상기 도 8의 (d) 또는 (e) 포맷을 사용하여 데이터를 재전송할 때 블록 끝 지시자인 END 필드를 다음 조건이 만족되면 '1'로 그렇지 않으면 '0'으로 설정한다.

<240>      첫째, 전송을 마친 후 더 이상 재전송하지 않은 데이터가 상기 블록의 데이터 조각에 남아 있지 않고.

<241>      둘째, 상기 D\_SEQ에 전송한 데이터 개수 N을 더한 값이  $N_{DATA}$  또는 그 배수 중 하나와 같지 않을 때 블록 끝 지시자인 END 필드를 '1'로 설정한다.

<242>      상기 RLP제어기 131은 다중화/역다중화 제어기 140이 16 비트, 20 비트 또는 32 비트의 길이를 가지는 데이터 블록을 요청하고, 보낼 새 데이터 또는 재전송할 데이터도 없고 전송할 제어 프레임도 없을 경우, 상기 도 8의 (f) 포맷과 같은 아이들 프레임을 전송할 수 있다. 상기 아이들 프레임을 만들기 위해서 상기 RLP제어기 131은 SEQ 필드를 L\_V(S) 레지스터 132의 상위 12 비트 값으로 채운다. 상기 RLP제어기 131은 나머지 데이터 블록 부분은 '0'으로 채운 데이터 블록을 상기 다중화/역다중화 제어기 140에게 전달

한다.

<243>      RLP제어기의 기본 채널 데이터 수신 동작

<244>      상기 다중화/역다중화 제어기 140은 상기 RLP제어기 131에게 데이터 블록과 함께 수신된 데이터 블록의 크기 T를 함께 알려준다.

<245>      상기 RLP제어기 131은 제어 프레임 중에서 SYNC, SYNC/ACK, ACK 프레임을 수신하면 상기 재설정 과정을 수행한다.

<246>      상기 RLP제어기 131은 제어 프레임 중에서 NAK 프레임을 수신하면 상기 <표 14>에 따라 NAK 프레임을 해석하여 요청된 블록이나 블록의 데이터 조각들을 재전송한다.

<247>      상기 RLP제어기 131은 상기 도 8의 (b), (c), (d), (e)와 같은 포맷의 새 데이터 프레임을 수신하였을 때 21 비트 일련 번호 L\_V(R) 레지스터 136을 사용한다. 상기 RLP 제어기 131은 상기 일련 번호 레지스터 L\_V(R) 136의 21 비트 값에서 상위 12 비트를 전송을 기다리는 블록 일련 번호 B\_SEQR로, 그리고 하위 9 비트를 전송을 기다리는 데이터 일련 번호 D\_SEQR로 결정한다. 우선 상기 RLP제어기 131은 수신된 프레임의 8 비트 SEQ 필드와 상기 L\_V(R) 레지스터 136을 통하여 다음과 같은 <수학식 2>를 통해 블록 일련 번호 B\_SEQ를 구한다.

<248>      **【수학식 2】**

$$B\_SEQ = [B\_SEQR + \{2^8 + SEQ - (B\_SEQR \text{ modulo } 2^8)\} \text{ modulo } 2^8] \text{ modulo } 2^{12}$$

- <249>      상기 RLP제어기 131은 상기 도 8의 (b) 또는 (c)와 같은 포맷의 새 데이터 프레임을 수신하였을 때 수신된 데이터 조각의 첫번째 데이터 바이트의 데이터 일련 번호 D\_SEQ를 0으로 결정한다. 상기 RLP제어기 131은 상기 도 8의 (b) 또는 (c)와 같은 포맷의 새 데이터 프레임을 수신하였을 때 블록 끝 지시자인 END 값을 '1'로 결정한다.
- <250>      상기 RLP제어기 131은 상기 도 8의 (d)와 같은 포맷의 새 데이터 프레임을 수신하였을 때 수신된 데이터 조각의 첫번째 데이터 바이트의 데이터 일련 번호 D\_SEQ를 수신된 프레임의 5 비트 SEG 필드와 상기 도 8의 (d)와 같은 포맷으로 보낼 수 있는 데이터 개수 B를 곱한 값으로 결정한다.
- <251>      상기 RLP제어기 131은 상기 도 8의 (e)와 같은 포맷의 새 데이터 프레임을 수신하였을 때 수신된 데이터 조각의 첫번째 데이터 바이트의 데이터 일련 번호 D\_SEQ를 수신된 프레임의 9 비트 SEG 필드값으로 결정한다.
- <252>      상기 RLP제어기 131은 상기 도 8의 (d) 또는 (e)와 같은 포맷의 새 데이터 프레임을 수신하였을 때 블록 끝 지시자인 END 값을 수신된 프레임의 1 비트 END 필드값으로 결정한다.
- <253>      상기 RLP제어기 131은 상기 도 8의 (b), (c), (d), (e)와 같은 포맷의 재전송 데이터 프레임을 수신하였을 때 상기 RLP제어기 131은 도 4에 도시된 바와 같이 NAK 리스트 137을 가지고 있어야 한다. 상기 NAK 리스트 137의 각 NAK 엔트리(NAK entry)들은 상기한 바와 같이 <빠진 데이터 바이트의 블록 일련 번호, 빠진 데이터 바이트의 데이터 일련 번호, 빠진 데이터 바이트가 수신되었는지를 알려주는 지시자, 재전송 타이머

(retransmit timer), 취소 타이머(abort timer)>로 구성된다. 상기 RLP제어기 131은 상기 NAK 리스트 137에 저장된 NAK 엔트리 중에서 12 비트 블록 일련 번호 값의 하위 8 비트 값과 상기 수신된 재전송 프레임의 8 비트 SEQ 필드에 담겨온 값이 일치하는 NAK 엔트리를 찾는다. 만일 같은 NAK 엔트리가 있다면, 상기 NAK 엔트리에 저장된 블록 일련 번호 B\_SEQ 값을 수신된 데이터 조각의 블록 일련 번호로 한다. 만일 그러한 엔트리가 없다면 상기 RLP제어기 131은 상기 수신된 프레임을 버린다.

<254>      상기 RLP제어기 131은 상기 도 9의 (a)와 같은 포맷의 재전송 데이터 프레임을 수신하였을 때 수신된 데이터 조각의 첫번째 데이터 바이트의 데이터 일련 번호 D\_SEQ를 0으로 결정한다.

<255>      상기 RLP제어기 131은 상기 도 8의 (b) 또는 (c)와 같은 포맷의 재전송 데이터 프레임을 수신하였을 때 블록 끝 지시자인 END 값을 '1'로 결정한다.

<256>      상기 RLP제어기 131은 상기 도 8의 (d)와 같은 포맷의 재전송 데이터 프레임을 수신하였을 때 수신된 데이터 조각의 첫번째 데이터 바이트의 데이터 일련 번호 D\_SEQ를 수신된 프레임의 5 비트 SEG 필드와 상기 도 8의 (d)와 같은 포맷으로 보낼 수 있는 데이터 개수 B를 곱한 값으로 결정한다.

<257>      상기 RLP제어기 131은 상기 도 8의 (e)와 같은 포맷의 재전송 데이터 프레임을 수신하였을 때 수신된 데이터 조각의 첫번째 데이터 바이트의 데이터 일련 번호 D\_SEQ를 수신된 프레임의 9 비트 SEG 필드값으로 결정한다.

<258>      상기 RLP제어기 131은 상기 도 8의 (d) 또는 (e)와 같은 포맷의 재전송 데이터 프레임을 수신하였을 때 블록 끝 지시자인 END 값을 수신된 프레임의 1 비트 END 필드값으로

로 결정한다.

<259> 상기 RLP제어기 131은 상기 도 8의 (b), (c), (d), (e) 포맷의 프레임을 수신하였을 때 수신된 데이터 조각의 첫번째 데이터 바이트의 데이터 일련 번호를 상기 과정에서 구한 D\_SEG 필드의 값으로 설정한다. 상기 RLP제어기 131은 수신된 데이터 조각의 첫번째 데이터 바이트의 블록 일련 번호 B\_SEQ와 데이터 일련 번호 D\_SEQ을 구하면 나머지 데이터 바이트의 일련 번호를 상기 <수학식 1>로 구할 수 있게 된다.

<260> 상기 RLP제어기 131은 수신된 데이터 조각의 각 데이터 바이트들을 처리할 때 21 비트 일련 번호 레지스터인 L\_V(N) 레지스터 136을 사용한다. 상기 RLP제어기 131은 상기 일련 번호 레지스터 L\_V(N) 135의 21 비트 값에서 상위 12 비트를 재전송을 기다리는 블록 일련 번호 B\_SEQN으로, 그리고 하위 9 비트를 재전송을 기다리는 데이터 일련 번호 D\_SEQN으로 결정한다. 상기 RLP제어기 131은 수신된 데이터 조각의 각 데이터 바이트들에 대해서 각 바이트가 가진 블록 일련 번호와 데이터 일련 번호를 다음 <표 17>에 나타난 바와 같이 판단한다.

<261> 【표 17】

경우	판단
경우 1	두 개의 블록 일련 번호, 데이터 일련 번호의 쌍 <B1, D1>, <B2, D2>가 있을 때, B1 = B2 이고 D1 = D2이면, <B1, D1>와 <B2, D2>는 같다고 한다.
경우 2	B2가 $(B1 + N_{BLOCK} / 2) \text{ modulo } N_{BLOCK}$ 에서 $(B1 + N_{BLOCK} - 1) \text{ modulo } N_{BLOCK}$ 까지의 일련 번호 중의 하나이면, <B1, D1>가 <B2, D2> 보다 크다고 한다.
경우 3	B1 = B2 이고 D2가 $(D1 + N_{DATA} / 2) \text{ modulo } N_{DATA}$ 에서 $(D1 + N_{DATA} - 1) \text{ modulo } N_{DATA}$ 까지의 일련 번호 중의 하나이면 <B1, D1>가 <B2, D2> 보다 크다고 한다.
경우 4	<B1, D1>이 <B2, D2>와 같지 않고 또한 크지도 않으면 <B1, D1>가 <B2, D2> 보다 작다고 한다.

<262> 상기 RLP제어기 131은 수신된 데이터 조각의 각 데이터 바이트들에 대해서 각 바이트



트가 가진 블록 일련 번호와 데이터 일련 번호 쌍 <B, D>를 상기 <B\_SEQR, D\_SEQR>, <B\_SEQN, D\_SEQN>와 <표 17>을 사용하여 다음과 같이 비교한다.

- <263> 첫째. <B, D>가 <B\_SEQN, D\_SEQN>보다 같거나 크고, <B\_SEQR, D\_SEQR>보다 작으면 상기 RLP제어기 131은 이를 재전송 데이터로 간주한다.
- <264> 둘째. <B, D>가 <B\_SEQR, D\_SEQR>보다 같거나 크고, B 값이 B\_SEQR와 E 레지스터 134를 더한 값을  $2^8$ 으로 modulo 연산한 값 보다 작으면 상기 RLP제어기 131은 이를 새 데이터로 간주한다.
- <265> 셋째. <B, D>가 <B\_SEQR, D\_SEQR>와 E 레지스터 134를 더한 값을  $2^8$ 으로 modulo 연산한 값 보다 같거나 크면 상기 RLP제어기 131은 이를 중복된 데이터로 간주한다.
- <266> 상기 RLP제어기 131은 상기에서 중복된 데이터로 간주된 데이터 바이트에 대해서는 이미 수신한 데이터이므로 버려버린다.
- <267> 상기 RLP제어기 131은 상기에서 수신된 데이터 바이트가 새 데이터 바이트 또는 재전송 데이터 바이트로 간주되었다면 상기 RLP제어기 131은 다음을 수행한다.
- <268> 첫째, 수신된 데이터 바이트의 <B, D>가 <B\_SEQN, D\_SEQN> 보다 크거나 같고 <B\_SEQR, D\_SEQR> 보다 작다면, 상기 RLP제어기 131은 상기한 바와 같이 <수신된 데이터 바이트의 블록 일련 번호, 수신된 데이터 바이트의 데이터 일련 번호, 수신된 데이터 바이트, 수신된 데이터 바이트가 블록의 마지막 데이터 바이트인지를 알려주는 지시자>로 구성된 새 재정렬 버퍼 엔트리를 만들어서 상기 수신된 데이터 바이트를 재정렬 버퍼 138에 저장한다. 상기 수신된 데이터 바이트가 블록의 마지막 데이터 바이트인지를 알려주는 지시자는 상기 데이터 바이트가 수신된 데이터 조각의 마지막 데이터 바이트이고 .

상기 END 값이 '1'로 설정되어 있으면 마지막임을 지시하도록 설정한다. 이때 상기 <B, D>가 <B\_SEQN, D\_SEQN>와 같다면, 상기 RLP제어기 131은 재정렬 버퍼 138에 저장된 재정렬 버퍼 엔트리들 중에서 일련 번호로 <B\_SEQN, D\_SEQN> 값을 갖는 재정렬 버퍼 엔트리에로부터 연속적으로 전달할 수 있는 일련 번호를 가지는 재정렬 버퍼 엔트리까지 저장된 데이터 바이트들을 순서대로 수신 데이터 버퍼에 전달한다. 이때 만일 전달된 마지막 데이터 바이트가 속한 재정렬 버퍼 엔트리의 수신된 데이터 바이트가 블록의 마지막 데이터 바이트임을 알려주는 지시자가 마지막 바이트임을 지시하는 값으로 설정되어 있으면 상기 RLP제어기 131은 상기 L\_V(N) 레지스터 135의 상위 12 비트를 상기 재정렬 버퍼 엔트리에 저장된 블록 일련 번호의 다음 블록 일련 번호로 설정하고 하위 9 비트를 0으로 설정한다. 만일 상기 과정에서 마지막 바이트임을 지시하지 않는 값으로 설정되어 있으면 상기 RLP제어기 131은 <B\_SEQN, D\_SEQN>를 다음 데이터 바이트를 가리키도록 상기 <수학식 1>을 사용하여 증가시킨 다음 상기 B\_SEQN 값과 D\_SEQN 값으로 상기 L\_V(N) 레지스터 135의 상위 12 비트와 하위 9 비트를 각각 설정한다. 상기 RLP제어기 131은 상기 수신된 데이터 바이트와 같은 블록 일련 번호, 데이터 일련 번호를 가지는 NAK 엔트리에 기다리는 데이터 프레임이 수신되었음을 기록한다.

<269> 둘째, 수신된 데이터 바이트의 <B, D>가 <B\_SEQR, D\_SEQR>과 같고, <B\_SEQR, D\_SEQR>가 <B\_SEQN, D\_SEQN>과 같은 경우, 상기 RLP제어기 131은 <B\_SEQR, D\_SEQR>과 <B\_SEQN, D\_SEQN>를 동시에 다음 데이터 바이트를 가리키도록 상기 <수학식 1>을 사용하여 증가시킨다. 이때 만일 상기 데이터 바이트가 수신된 데이터 조각의 마지막 데이터 바이트이고 상기 END 값이 '1'로 설정되어 있으면 상기 RLP제어기 131은 상기 B\_SEQN 값의 다음 블록 일련 번호로 상기 L\_V(N) 레지스터 135의 상위 12 비트를 설정하고 하위 9

비트는 0으로 설정한 다음 마찬가지로 상기 B\_SEQR 값의 다음 블록 일련 번호로 상기 L\_V(R) 레지스터 136의 상위 12 비트를 설정하고 하위 9 비트는 0으로 설정한다. 만일 그렇지 않으면 상기 RLP제어기 131은 상기 B\_SEQN 값과 D\_SEQN 값으로 상기 L\_V(N) 레지스터 135의 상위 12 비트와 하위 9 비트를 각각 설정하고 상기 B\_SEQR 값과 D\_SEQR 값으로 상기 L\_V(R) 레지스터 136의 상위 12 비트와 하위 9 비트를 각각 설정한다. 상기 RLP 제어기 131은 상기 수신된 데이터 바이트를 수신 데이터 버퍼에 전달한다. 이와 달리 수신된 데이터 바이트의 <B, D>가 <B\_SEQR, D\_SEQR>과 같고, <B\_SEQR, D\_SEQR>가 <B\_SEQN, D\_SEQN>과 같지 않은 경우, <B\_SEQR, D\_SEQR>를 다음 데이터 바이트를 가리키도록 상기 <수학식 1>을 사용하여 증가시킨다. 이때 만일 상기 데이터 바이트가 수신된 데이터 조각의 마지막 데이터 바이트이고 상기 END 값이 '1'로 설정되어 있으면 상기 RLP 제어기 131은 상기 B\_SEQR 값의 다음 블록 일련 번호로 상기 L\_V(R) 레지스터 136의 상위 12 비트를 설정하고 하위 9 비트는 0으로 설정한다. 만일 그렇지 않으면 상기 RLP제어기 131은 상기 B\_SEQR 값과 D\_SEQR 값으로 상기 L\_V(R) 레지스터 136의 상위 12 비트와 하위 9 비트를 각각 설정한다. 이 경우, 상기 RLP제어기 131은 상기 수신된 데이터 바이트를 상기한 바와 같이 <수신된 데이터 바이트의 블록 일련 번호, 수신된 데이터 바이트의 데이터 일련 번호, 수신된 데이터 바이트, 수신된 데이터 바이트가 블록의 마지막 데이터 바이트인지를 알려주는 지시자>로 구성된 새 재정렬 버퍼 엔트리를 만들어서 상기 수신된 데이터 바이트를 재정렬 버퍼 138에 저장한다. 상기 수신된 데이터 바이트가 블록의 마지막 데이터 바이트인지를 알려주는 지시자는 상기 데이터 바이트가 수신된 데이터 조각의 마지막 데이터 바이트이고 상기 END 값이 '1'로 설정되어 있으면 마지막임을 지시하도록 설정한다.

<270> 셋째, 수신된 데이터 바이트의 <B, D>가 <B\_SEQR, D\_SEQR> 보다 크다면 상기 RLP 제어기 131은 <B\_SEQR, D\_SEQR>에서 <B, D>의 바로 전 데이터 바이트에 대한 재전송을 요청하기 위해 NAK 리스트 137에 각 데이터 바이트에 대한 엔트리를 만든다. 각 엔트리는 상기한 바와 같이 <빠진 데이터 바이트의 블록 일련 번호, 빠진 데이터 바이트의 데이터 일련 번호, 빠진 데이터 바이트가 수신되었는지를 알려주는 지시자, 재전송 타이머(retransmit timer), 취소 타이머(abort timer)>로 구성된다. 또한, 상기 RLP제어기 131은 수신된 데이터 바이트를 재정렬 버퍼 138에 저장하고 <B\_SEQR, D\_SEQR>를 다음 데이터 바이트를 가리키도록 상기 <수학식 1>을 사용하여 증가시킨다. 이때 만일 상기 데이터 바이트가 수신된 데이터 조각의 마지막 데이터 바이트이고 상기 END 값이 '1'로 설정되어 있으면 상기 RLP제어기 131은 상기 B\_SEQR 값의 다음 블록 일련 번호로 상기 L\_V(R) 레지스터 136의 상위 12 비트를 설정하고 하위 9 비트는 0으로 설정한다. 만일 그렇지 않으면 상기 RLP제어기 131은 상기 B\_SEQR 값과 D\_SEQR 값으로 상기 L\_V(R) 레지스터 136의 상위 12 비트와 하위 9 비트를 각각 설정한다.

<271> 한편, 상기 RLP제어기 131은 상기 도 8의 (f)와 같은 포맷의 아이들 프레임을 받으면, 블록 일련 번호 B\_SEQ를 12 비트 SEQ 필드 값으로 설정하고 데이터 일련 번호 D\_SEQ를 0으로 설정한다. 상기 RLP제어기 131은 상기 수신된 아이들 프레임의 <B\_SEQ, D\_SEQ>가 상기 <B\_SEQR, D\_SEQR> 보다 크면, 상기 RLP제어기 131은 <B\_SEQR, D\_SEQR>에서 <B, D>의 바로 전 데이터 바이트에 대한 재전송을 요청하기 위해 NAK 리스트 137에 각 데이터 바이트에 대한 엔트리를 만든다. 각 엔트리는 상기한 바와 같이 <빠진 데이터 바이트의 블록 일련 번호, 빠진 데이터 바이트의 데이터 일련 번호, 빠진 데이터 바이트가

수신되었는지를 알려주는 지시자, 재전송 타이머(retransmit timer), 취소 타이머(abort timer)>로 구성된다. 또한, 상기 RLP제어기 131은 상기 B\_SEQR 값의 다음 블록 일련 번호로 상기 L\_V(R) 레지스터 136의 상위 12 비트를 설정하고 하위 9 비트는 0으로 설정한다. 상기 RLP제어기 131은 상기 수신된 아이들 프레임의 <B, D>가 <B\_SEQR, D\_SEQR> 보다 작으면 상기 재설정 과정을 수행한다.

<272>      상기 RLP제어기 131은 상기 다중화/역다중화 제어기가 깨진 데이터 블록이 수신되었음을 알리면 레지스터 E 134에 저장된 값을 하나 증가시켜 다시 레지스터 E 134에 저장한다. 이때 상기 증가된 E 레지스터 134와 B\_SEQR을 더하여  $2^8$  modulo 연산을 한 결과가 상기 B\_SEQN 보다 크게 되면 상기 RLP제어기 131은 상기 재설정 과정을 수행한다.

<273>      상기 RLP제어기 131은 데이터 수신 과정에서 빈 데이터 블록이 아니면서 올바르게 수신된 데이터 블록이 하나라도 있거나 상기 다중화/역다중화 제어기가 아무런 프레임도 수신되지 않았다고 알려주면 상기 레지스터 E 134를 0으로 설정한다.

#### <274>      RLP제어기의 부가 채널 데이터 송신 동작

<275>      상기 송신측 다중화/역다중화 제어기 140은 상기 RLP제어기 131에게 부가 채널로 전송할 데이터 블록을 요청하여 상기 RLP제어기 131이 프레임을 생성하도록 한다. 상기 송신측 다중화/역다중화 제어기 140은 상기 RLP제어기 131에게 데이터 블록을 요청하면서 상기 RLP제어기 131이 만들어내야하는 데이터 블록의 크기 T를 함께 알려준다.

<276>      상기 RLP제어기 131은 부가 채널로 전송할 데이터 블록을 만들 때 다음의 우선 순

위에 따라 데이터 블록을 만든다.

<277> 첫째, 재전송 데이터 프레임

<278> 둘째, 새 데이터 프레임

<279> RLP제어기 131이 부가 채널로 전송할 새 데이터 프레임을 생성할 때는 다음과 같이 동작한다.

<280> 상기 RLP제어기 131은 새 데이터 프레임을 가장 처음 생성할 때 또는 이전에 정의 하였던 블록이 가지고 있는 모든 데이터 바이트를 전송하였으면 데이터열의 남은 데이터 에 대해서 새 블록을 정의한다. 본 발명에서는 새 블록을 정의할 때 상기 기본 채널에서 와 마찬가지로 상기 다중화/역다중화 제어기 140이 요청한 데이터 블록에 담을 수 있는 데이터 크기만큼 블록이 데이터 조각을 가지도록 정의하는 방식을 모드 A(mode A)라고 부르고, 반면 상기 다중화/역다중화 제어기 140이 요청한 데이터 블록의 크기에 관계없 이 언제나  $N_{DATA}$  바이트의 데이터를 가지도록 정의하는 방식을 모드 B(mode B)라고 부르 겠다. 상기 RLP제어기 131은 새 블록을 정의할 때마다 모드 A와 모드 B 중에서 가장 적 절한 방식을 선택할 수 있다.

<281> 도 9에서는 모드 A와 B에서 부가 채널로 전송할 수 있는 사용가능한 프레임 포맷들 을 보여주고 있다. 상기 모드 A에서 생성된 새 데이터 프레임이 가질 수 있는 포맷이 상 기 도 9의 (a) 포맷이다. 상기 모드 B에서 생성된 새 데이터 프레임이 가질 수 있는 포 맷은 상기 도 9의 (b) 포맷이다. 상기 RLP제어기 131은 새 데이터를 담은 부가 채널 RLP 프레임을 생성할 때 21 비트 일련 번호 레지스터 L\_V(S) 132를 사용한다. 상기 RLP제어

기 131은 상기 일련 번호 레지스터 L\_V(S) 132의 21 비트 값에서 상위 12 비트를 블록 일련 번호 B\_SEQ로, 그리고 하위 9 비트를 데이터 일련 번호 D\_SEQ로 결정한다.

<282>      상기 RLP제어기 131은 상기 과정에서 구해진 D\_SEQ 값이 0이 아니면 아직 전송하지 않은 블록의 데이터 부분이 있는 것으로 판단한다. 즉, 상기 블록 일련 번호 B\_SEQ에 해당하는 블록의 데이터 일련 번호 D\_SEQ에서부터 ( $N_{DATA} - 1$ )까지의 데이터 바이트들이 전송되어야 한다고 판단한다. 따라서 상기 RLP제어기 131은 상기 데이터 조각을 전송하기 위해서 모드 B에서 사용할 수 있는 프레임 포맷인 도 9의 (b) 포맷을 사용한다.

<283>      상기 RLP제어기 131은 상기 도 9의 (b) 포맷을 사용하여 새 데이터를 전송할 때 먼저 SEQ 필드를 상기 데이터 부분의 블록 일련 번호인 B\_SEQ 값의 하위 8 비트로 설정한다. 상기 RLP제어기 131은 또한 상기 D\_SEQ 값으로 SEG 필드를 설정한다.

<284>      상기 RLP제어기 131은 상기 도 9의 (b) 포맷을 사용하여 새 데이터를 전송할 때 TYPE 필드를 상기 도 8과 같이 '1'로, 재전송 지시자인 REXMIT 필드를 '0'으로 설정한 다음, DATA 필드를 전송할 데이터로 채운다.

<285>      상기 RLP제어기 131은 상기 도 9의 (b) 포맷을 사용하여 새 데이터를 전송할 때 블록 끝 지시자인 END 필드를 다음 조건이 만족되면 '1'로 그렇지 않으면 '0'으로 설정한다.

<286>      첫째, 전송을 마친 후 더 이상 전송하지 않은 데이터가 데이터열에 남아 있지 않고.

<287>      둘째, 상기 D\_SEQ에 전송한 데이터 개수 N을 더한 값이  $N_{DATA}$  또는 그 배수 중 하나와 같지 않을 때 블록 끝 지시자인 END 필드를 '1'로 설정한다.

- <288>      상기 RLP제어기 131은 상기 과정에서 새 데이터 프레임을 전송하면 다음과 같이 L\_V(S) 레지스터 132를 수정한다. 상기 RLP제어기 131은 상기 과정에서 전송된 새 데이터의 크기가 N이면 상기 L\_V(S) 레지스터 132에 N 값을 더한다.
- <289>      상기 RLP제어기 131은 상기 D\_SEQ 값이 0이면 새 블록을 정의한다. 상기한 바와 같이 이 작업은 모드 A일 때와 모드 B일 때 수행하는 절차가 달라지게 된다. 만일 상기 RLP제어기 131이 모드 A에 따라서 새 블록을 정의하게 되면 상기 RLP제어기 131은 다음과 같이 동작한다.
- <290>      상기 RLP제어기 131은 모드 A에서 새 블록의 크기를 상기 도 9의 (a)로 보낼 수 있는 데이터 개수 B, 즉 상기 다중화/역다중화 제어기 140이 요청한 데이터 블록의 크기 T에서 RLP 헤더인 10 비트를 뺀 값과 아직 전송되지 않은 데이터열의 데이터 수 중에서 최소값 M으로 정한다.
- <291>      상기 RLP제어기 131은 상기 도 9의 (a) 포맷을 사용하여 새 데이터를 전송할 때 먼저 SEQ 필드를 상기 데이터 부분의 블록 일련 번호인 B\_SEQ 값의 하위 8 비트로 설정한다. 상기 도 9의 (a) 포맷으로 전달되는 데이터는 언제나 데이터 일련 번호가 0부터 시작하고 그 크기가 N<sub>DATA</sub> 보다 작으므로 블록 끝 지시자도 언제나 '1'인 것에 주의하여야 한다. 따라서 상기 도 9의 (a) 포맷에서는 상기 데이터 일련 번호와 블록 끝 지시자를 포함하지 않았지만 수신측 RLP제어기는 상기 도 9의 (a) 포맷을 수신하면 데이터 일련 번호를 0으로, 블록 끝 지시자를 '1'로 인식한다.
- <292>      상기 RLP제어기 131은 상기 도 9의 (a) 포맷을 사용하여 새 데이터를 전송할 때



TYPE 필드를 상기 도 8과 같이 '0'로, 재전송 지시자인 REXMIT 필드를 '0'으로 설정한 다음, DATA 필드를 전송할 데이터로 채운다.

<293>      상기 RLP제어기 131은 상기 과정에서 새 데이터 프레임을 전송하면 다음과 같이 L\_V(S) 레지스터 132를 수정한다. 상기 RLP제어기 131은 상기 L\_V(S) 레지스터 132에 NDATA 값을 더한다.

<294>      만일 RLP제어기 131이 모드 B에 따라서 새 블록을 정의하게 되면 상기 RLP제어기 131은 다음과 같이 동작한다. 상기 RLP제어기 131은 모드 B에서 새 블록의 크기를 NDATA 바이트와 아직 전송되지 않은 데이터열의 데이터 수 중에서 최소값 M으로 정한다. 상기 RLP제어기 131은 새 블록의 크기를 정하면 아직 전송되지 않은 데이터열의 데이터에서 정해진 블록 크기 만큼의 데이터를 새 블록이 가지고 있는 것으로 정의한다. 상기 RLP제어기 131은 상기와 같이 새 블록을 정의하면 정의된 블록에 대해서 상기 블록의 남은 부분을 전송하는 절차를 수행한다.

<295>      상기 RLP제어기 131은 새로 정의한 블록들을 기본 채널과 마찬가지로 수신측의 재전송 요청에 대비하여 재전송 버퍼 133에 저장해둔다.

<296>      상기 RLP제어기 131은 재전송 데이터 프레임을 보낼 때는 상기 도 9의 (a) 또는 (b) 포맷을 사용할 수 있다. 상기 RLP제어기 131은 상기 도 9의 (a) 포맷을 다음 조건이 만족될 때 사용한다.

<297>      첫째, 재전송 요청이 블록 전체에 대한 것이고,

<298>      둘째, 재전송될 블록의 크기가 상기 도 9의 (a) 포맷으로 보낼 수 있는 데이터 개수 B, 즉 상기 다중화/역다중화 제어기 140이 요청한 데이터 블록의 크기 T에서 RLP 헤

더인 10 비트를 뺀 값과 같거나 작은 경우 상기 RLP제어기 131은 상기 도 9의 (a) 포맷을 사용한다.

<299>       상기 RLP제어기 131은 상기 도 9의 (a) 포맷을 사용하여 상기 블록을 재전송할 때 먼저 SEQ 필드를 상기 데이터 부분의 블록 일련 번호인 B\_SEQ 값의 하위 8 비트로 설정한다. 상기 RLP제어기 131은 상기 도 9의 (a) 포맷을 사용하여 상기 블록을 재전송할 때 TYPE 필드를 상기 도 9와 같이 '0'로, 재전송 지시자인 REXMIT 필드를 '1'로 설정한 다음, DATA 필드를 재전송할 데이터로 채운다. 상기 RLP제어기 131은 상기 조건을 만족시키지 못하면 상기 도 9의 (b) 포맷을 사용한다.

<300> ...     상기 RLP제어기 131은 상기 도 9의 (b) 포맷을 사용하여 상기 블록의 데이터 조각을 재전송할 때 먼저 SEQ 필드를 상기 데이터 부분의 블록 일련 번호인 B\_SEQ 값의 하위 8 비트로 설정한다. 상기 RLP제어기 131은 또한 상기 D\_SEQ 값을 상기 도 9의 (b) 포맷으로 보내는 데이터 개수 B로 나눈 몫으로 SEG 필드를 설정한다.

<301>       상기 RLP제어기 131은 상기 도 9의 (b) 포맷을 사용하여 상기 블록의 데이터 조각을 재전송할 때 TYPE 필드를 상기 도 8과 같이 '1'로, 재전송 지시자인 REXMIT 필드를 '1'으로 설정한 다음, DATA 필드를 전송할 데이터로 채운다.

<302>       상기 RLP제어기 131은 상기 도 9의 (b) 포맷을 사용하여 데이터를 재전송할 때 블록 끝 지시자인 END 필드를 다음 조건이 만족되면 '1'로 그렇지 않으면 '0'으로 설정한다.

<303>       첫째, 전송을 마친 후 더 이상 재전송하지 않은 데이터가 상기 블록의 데이터 조각

에 남아 있지 않고.

<304> 둘째, 상기 D\_SEQ에 전송한 데이터 개수 N를 더한 값이 N<sub>DATA</sub> 또는 그 배수 중 하나와 같지 않을 때 블록 끝 지시자인 END 필드를 '1'로 설정한다.

<305> 상기 RLP제어기 131은 보낼 새 데이터 또는 재전송할 데이터도 없고 전송할 제어 프레임도 없을 경우, 빈 데이터 블록을 상기 다중화/역다중화 제어기 140에게 전달한다.

<306> RLP제어기의 부가 채널 데이터 수신 동작

<307> 상기 다중화/역다중화 제어기 140은 상기 RLP제어기 131에게 데이터 블록과 함께 수신된 데이터 블록의 크기 T를 함께 알려준다.

<308> 상기 RLP제어기 131은 상기 도 9의 (a) 또는 (b)와 같은 포맷의 새 데이터 프레임을 수신하였을 때 21 비트 일련 번호 L\_V(R) 레지스터 136을 사용한다. 상기 RLP제어기 131은 상기 일련 번호 레지스터 L\_V(R) 136의 21 비트 값에서 상위 12 비트를 전송을 기다리는 블록 일련 번호 B\_SEQR로, 그리고 하위 9 비트를 전송을 기다리는 데이터 일련 번호 D\_SEQR로 결정한다. 우선 상기 RLP제어기 131은 수신된 프레임의 8 비트 SEQ 필드와 상기 L\_V(R) 레지스터 136을 통하여 상기 <수학식 2>를 통해 블록 일련 번호 B\_SEQ를 구한다.

<309> 상기 RLP제어기 131은 상기 도 9의 (a)와 같은 포맷의 새 데이터 프레임을 수신하였을 때 수신된 데이터 조각의 첫번째 데이터 바이트의 데이터 일련 번호 D\_SEQ를 0으로 결정한다.

- <310>      상기 RLP제어기 131은 상기 도 9의 (a)와 같은 포맷의 새 데이터 프레임을 수신하였을 때 블록 끝 지시자인 END 값을 '1'로 결정한다.
- <311>      상기 RLP제어기 131은 상기 도 9의 (b)와 같은 포맷의 새 데이터 프레임을 수신하였을 때 수신된 데이터 조각의 첫번째 데이터 바이트의 데이터 일련 번호 D\_SEQ를 수신된 프레임의 9 비트 SEG 필드값으로 결정한다. 상기 RLP제어기 131은 상기 도 9의 (b)와 같은 포맷의 새 데이터 프레임을 수신하였을 때 블록 끝 지시자인 END 값을 수신된 프레임의 1 비트 END 필드값으로 결정한다.
- <312>      상기 RLP제어기 131은 상기 도 9의 (a) 또는 (b)와 같은 포맷의 재전송 데이터 프레임을 수신하였을 때 상기 RLP제어기 131은 상기 기본 채널에서와 같이 NAK 리스트 137을 가지고 있어야 한다. 상기 RLP제어기 131은 상기 NAK 리스트 137에 저장된 NAK 엔트리 중에서 12 비트 블록 일련 번호 값의 하위 8 비트 값과 상기 수신된 재전송 프레임의 8 비트 SEQ 필드에 담겨온 값이 일치하는 NAK 엔트리를 찾는다. 만일 같은 NAK 엔트리가 있다면, 상기 NAK 엔트리에 저장된 블록 일련 번호 B\_SEQ 값을 수신된 데이터 조각의 블록 일련 번호로 한다. 만일 그러한 엔트리가 없다면 상기 RLP제어기 131은 상기 수신된 프레임을 버린다.
- <313>      상기 RLP제어기 131은 상기 도 9의 (a)와 같은 포맷의 재전송 데이터 프레임을 수신하였을 때 수신된 데이터 조각의 첫번째 데이터 바이트의 데이터 일련 번호 D\_SEQ를 0으로 결정한다. 상기 RLP제어기 131은 상기 도 9의 (a)와 같은 포맷의 재전송 데이터 프레임을 수신하였을 때 블록 끝 지시자인 END 값을 '1'로 결정한다.
- <314>      상기 RLP제어기 131은 상기 도 9의 (b)와 같은 포맷의 재전송 데이터 프레임을 수신하였을 때 수신된 데이터 조각의 첫번째 데이터 바이트의 데이터 일련 번호 D\_SEQ를

수신된 프레임의 9 비트 SEG 필드값으로 결정한다. 상기 RLP제어기 131은 상기 도 9의 (b)와 같은 포맷의 재전송 데이터 프레임을 수신하였을 때 블록 끝 지시자인 END 값을 수신된 프레임의 1 비트 END 필드값으로 결정한다.

- <315>      상기 RLP제어기 131은 상기 도 9의 (a) 또는 (b) 포맷의 프레임을 수신하였을 때 수신된 데이터 조각의 첫번째 데이터 바이트의 데이터 일련 번호를 상기 과정에서 구한 D\_SEG 필드의 값으로 설정한다. 상기 RLP제어기 131은 수신된 데이터 조각의 첫번째 데이터 바이트의 블록 일련 번호 B\_SEQ와 데이터 일련 번호 D\_SEQ을 구하면 나머지 데이터 바이트의 일련 번호를 상기 <수학식 1>로 구할 수 있게 된다.
- <316>      상기 RLP제어기 131은 수신된 데이터 조각의 각 데이터 바이트들을 처리할 때 21 비트 일련 번호 레지스터인 L\_V(N) 레지스터 135를 사용한다. 상기 RLP제어기 131은 상기 일련 번호 레지스터 L\_V(N) 135의 21 비트 값에서 상위 12 비트를 재전송을 기다리는 블록 일련 번호 B\_SEQN으로, 그리고 하위 9 비트를 재전송을 기다리는 데이터 일련 번호 D\_SEQN으로 결정한다. 상기 RLP제어기 131은 수신된 데이터 조각의 각 데이터 바이트들에 대해서 각 바이트가 가진 블록 일련 번호와 데이터 일련 번호를 상기 <표 17>을 사용하여 비교한다.
- <317>      상기 RLP제어기 131은 수신된 데이터 조각의 각 데이터 바이트들에 대해서 각 바이트가 가진 블록 일련 번호와 데이터 일련 번호 쌍 <B, D>를 상기 <B\_SEQR, D\_SEQR>, <B\_SEQN, D\_SEQN>와 <표 17>을 사용하여 다음과 같이 비교한다.
- <318>      첫째. <B, D>가 <B\_SEQN, D\_SEQN>보다 같거나 크고, <B\_SEQR, D\_SEQR>보다 작으면 상기 RLP제어기 131은 이를 재전송 데이터로 간주한다.

- <319> 둘째. <B, D>가 <B\_SEQR, D\_SEQR>보다 같거나 크고, B 값이 B\_SEQR와 E 레지스터 134를 더한 값을  $2^8$ 으로 modulo 연산한 값 보다 작으면 상기 RLP제어기 131은 이를 새 데이터로 간주한다.
- <320> 셋째. <B, D>가 <B\_SEQR, D\_SEQR>와 E 레지스터 134를 더한 값을  $2^8$ 으로 modulo 연산한 값 보다 같거나 크면 상기 RLP제어기 131은 이를 중복된 데이터로 간주한다.
- <321> 상기 RLP제어기 131은 상기에서 중복된 데이터로 간주된 데이터 바이트에 대해서는 이미 수신한 데이터이므로 버려버린다.
- <322> 상기 RLP제어기 131은 상기에서 수신된 데이터 바이트가 새 데이터 바이트 또는 재 전송 데이터 바이트로 간주되었다면 상기 RLP제어기 131은 다음을 수행한다.
- <323> 첫째, 수신된 데이터 바이트의 <B, D>가 <B\_SEQN, D\_SEQN> 보다 크거나 같고 <B\_SEQR, D\_SEQR> 보다 작다면, 상기 RLP제어기 131은 상기한 바와 같이 <수신된 데이터 바이트의 블록 일련 번호, 수신된 데이터 바이트의 데이터 일련 번호, 수신된 데이터 바이트, 수신된 데이터 바이트가 블록의 마지막 데이터 바이트인지를 알려주는 지시자>로 구성된 새 재정렬 버퍼 엔트리를 만들어서 상기 수신된 데이터 바이트를 재정렬 버퍼 138에 저장한다. 상기 수신된 데이터 바이트가 블록의 마지막 데이터 바이트인지를 알려주는 지시자는 상기 데이터 바이트가 수신된 데이터 조각의 마지막 데이터 바이트이고 상기 END 값이 '1'로 설정되어 있으면 마지막임을 지시하도록 설정한다. 이때 상기 <B, D>가 <B\_SEQN, D\_SEQN>와 같다면, 상기 RLP제어기 131은 재정렬 버퍼 138에 저장된 재정렬 버퍼 엔트리들 중에서 일련 번호로 <B\_SEQN, D\_SEQN> 값을 갖는 재정렬 버퍼 엔트리

에서부터 연속적으로 전달할 수 있는 일련 번호를 가지는 재정렬 버퍼 엔트리까지 저장된 데이터 바이트들을 순서대로 수신 데이터 버퍼에 전달한다. 이때 만일 전달된 마지막 데이터 바이트가 속한 재정렬 버퍼 엔트리의 수신된 데이터 바이트가 블록의 마지막 데이터 바이트인지를 알려주는 지시자가 마지막 바이트임을 지시하는 값으로 설정되어 있으면 상기 RLP제어기 131은 상기 L\_V(N) 레지스터 135의 상위 12 비트를 상기 재정렬 버퍼 엔트리에 저장된 블록 일련 번호의 다음 블록 일련 번호로 설정하고 하위 9 비트를 0으로 설정한다. 만일 상기 과정에서 마지막 바이트임을 지시하지 않는 값으로 설정되어 있으면 상기 RLP제어기 131은 <B\_SEQN, D\_SEQN>를 다음 데이터 바이트를 가리키도록 상기 <수학식 1>을 사용하여 증가시킨 다음 상기 B\_SEQN 값과 D\_SEQN 값으로 상기 L\_V(N) 레지스터 135의 상위 12 비트와 하위 9 비트를 각각 설정한다. 상기 RLP제어기 131은 상기 수신된 데이터 바이트와 같은 블록 일련 번호, 데이터 일련 번호를 가지는 NAK 엔트리에 기다리는 데이터 프레임이 수신되었음을 기록한다.

<324> 둘째, 수신된 데이터 바이트의 <B, D>가 <B\_SEQR, D\_SEQR>과 같고, <B\_SEQR, D\_SEQR>가 <B\_SEQN, D\_SEQN>과 같은 경우, 상기 RLP제어기 131은 <B\_SEQR, D\_SEQR>과 <B\_SEQN, D\_SEQN>를 동시에 다음 데이터 바이트를 가리키도록 상기 <수학식 1>을 사용하여 증가시킨다. 이때 만일 상기 데이터 바이트가 수신된 데이터 조각의 마지막 데이터 바이트이고 상기 END 값이 '1'로 설정되어 있으면 상기 RLP제어기 131은 상기 B\_SEQN 값의 다음 블록 일련 번호로 상기 L\_V(N) 레지스터 135의 상위 12 비트를 설정하고 하위 9 비트는 0으로 설정한 다음 마찬가지로 상기 B\_SEQR 값의 다음 블록 일련 번호로 상기 L\_V(R) 레지스터 136의 상위 12 비트를 설정하고 하위 9 비트는 0으로 설정한다. 만일 그렇지 않으면 상기 RLP제어기 131은 상기

B\_SEQN 값과 D\_SEQN 값으로 상기 L\_V(N) 레지스터 135의 상위 12 비트와 하위 9 비트를 각각 설정하고 상기 B\_SEQR 값과 D\_SEQR 값으로 상기 L\_V(R) 레지스터 136의 상위 12 비트와 하위 9 비트를 각각 설정한다. 상기 RLP제어기 131은 상기 수신된 데이터 바이트를 수신 데이터 버퍼에 전달한다. 이와 달리 수신된 데이터 바이트의 <B, D>가 <B\_SEQR, D\_SEQR>과 같고, <B\_SEQR, D\_SEQR>가 <B\_SEQN, D\_SEQN>과 같지 않은 경우, <B\_SEQR, D\_SEQR>를 다음 데이터 바이트를 가리키도록 상기 <수학식 1>을 사용하여 증가시킨다. 이때 만일 상기 데이터 바이트가 수신된 데이터 조각의 마지막 데이터 바이트이고 상기 END 값이 '1'로 설정되어 있으면 상기 RLP제어기 131은 상기 B\_SEQR 값의 다음 블록 일련 번호로 상기 L\_V(R) 레지스터 136의 상위 12 비트를 설정하고 하위 9 비트는 0으로 설정한다. 만일 그렇지 않으면 상기 RLP제어기 131은 상기 B\_SEQR 값과 D\_SEQR 값으로 상기 L\_V(R) 레지스터 136의 상위 12 비트와 하위 9 비트를 각각 설정한다. 이 경우, 상기 RLP제어기 131은 상기 수신된 데이터 바이트를 상기한 바와 같이 <수신된 데이터 바이트의 블록 일련 번호, 수신된 데이터 바이트의 데이터 일련 번호, 수신된 데이터 바이트, 수신된 데이터 바이트가 블록의 마지막 데이터 바이트인지를 알려주는 지시자>로 구성된 새 재정렬 버퍼 엔트리를 만들어서 상기 수신된 데이터 바이트를 재정렬 버퍼 138에 저장한다. 상기 수신된 데이터 바이트가 블록의 마지막 데이터 바이트인지를 알려주는 지시자는 상기 데이터 바이트가 수신된 데이터 조각의 마지막 데이터 바이트이고 상기 END 값이 '1'로 설정되어 있으면 마지막임을 지시하도록 설정한다.

<325> 셋째, 수신된 데이터 바이트의 <B, D>가 <B\_SEQR, D\_SEQR> 보다 크다면 상기



RLP제어기 131은 <B\_SEQR, D\_SEQR>에서 <B, D>의 바로 전 데이터 바이트에 대한 재전송을 요청하기 위해 NAK 리스트 137에 각 데이터 바이트에 대한 엔트리를 만든다. 각 엔트리는 상기한 바와 같이 <빠진 데이터 바이트의 블록 일련 번호, 빠진 데이터 바이트의 데이터 일련 번호, 빠진 데이터 바이트가 수신되었는지를 알려주는 지시자, 재전송 타이머(retransmit timer), 취소 타이머(abort timer)>로 구성된다. 또한, 상기 RLP제어기 131은 수신된 데이터 바이트를 재정렬 버퍼 138에 저장하고 <B\_SEQR, D\_SEQR>를 다음 데이터 바이트를 가리키도록 상기 <수학식 1>을 사용하여 증가시킨다. 이때 만일 상기 데이터 바이트가 수신된 데이터 조각의 마지막 데이터 바이트이고 상기 END 값이 '1'로 설정되어 있으면 상기 RLP제어기 131은 상기 B\_SEQR 값의 다음 블록 일련 번호로 상기 L\_V(R) 레지스터 136의 상위 12 비트를 설정하고 하위 9 비트는 0으로 설정한다. 만일 그렇지 않으면 상기 RLP제어기 131은 상기 B\_SEQR 값과 D\_SEQR 값으로 상기 L\_V(R) 레지스터 136의 상위 12 비트와 하위 9 비트를 각각 설정한다.

<326>      상기 RLP제어기 131은 상기 다중화/역다중화 제어기가 깨진 데이터 블록이 수신되었음을 알리면 레지스터 E 134에 저장된 값을 하나 증가시켜 다시 레지스터 E 134에 저장한다. 이때 상기 증가된 E 레지스터 134와 B\_SEQR을 더하여  $2^8 \text{ modulo}$  연산을 한 결과가 상기 B\_SEQN 보다 크게 되면 상기 RLP제어기 131은 상기 재설정 과정을 수행한다.

<327>      상기 RLP제어기 131은 데이터 수신 과정에서 빈 데이터 블록이 아니면서 올바르게 수신된 데이터 블록이 하나라도 있거나 상기 다중화/역다중화 제어기가 아무런 프레임도 수신되지 않았다고 알려주면 상기 레지스터 E 134를 0으로 설정한다.

<328>      RLP제어기의 데이터 수신후 동작

<329>      RLP제어기 131은 수신된 모든 프레임을 처리한 후 다음과 같은 작업을 수행한다.

<330>      수신된 데이터 블록 중에서 새 데이터 프레임이 수신되거나 송신측 RLP가 아무런 프레임도 보내지 않았다고 판단되는 경우, 상기 RLP제어기 131은 NAK 리스트 137에서 오래된 NAK 엔트리부터 다음의 과정을 수행한다.

<331>      첫째, NAK 엔트리가 취소 타이머가 설정된 후 아직 만료되지 않았고 이미 세 번의 NAK에 상기 NAK 엔트리가 가지고 있는 일련 번호가 포함되어 전송되었다면, RLP제어기 131은 취소 타이머의 값을 하나 줄인다. 만일 상기 취소 타이머 값이 0이 되면, RLP제어기 131은 다음 동작을 수행한다. 만일 RLP제어기 131이 이미 NAK 엔트리가 가지고 있는 일련 번호에 해당하는 재전송 데이터 바이트를 수신하였다면, 상기 RLP제어기 131은 상기 NAK 엔트리를 삭제한다. 이와 달리, RLP제어기 131이 NAK 엔트리가 가지고 있는 일련 번호에 해당하는 재전송 데이터 바이트를 수신하지 못했다면, 상기 RLP제어기 131은 상기 NAK 엔트리의 일련 번호에 해당하는 재전송 데이터 바이트를 수신하지 못한 것으로 간주하고, 재정렬 버퍼 138에 저장된 상기 NAK 엔트리의 블록 일련 번호, 데이터 일련 번호 보다 크고 연속적으로 상위 링크 프로토콜로 보낼 수 있는 수신된 데이터 바이트들을 순서대로 수신 데이터 버퍼에 전달한다. 상기 RLP제어기 131은 다음번 수신을 기다리는 데이터 바이트의 일련 번호로 L\_V(N) 레지스터 135를 설정한다.

<332>      둘째, NAK 엔트리가 재전송 타이머가 설정된 후 아직 만료되지 않았고 이미 두 번의 NAK에 상기 NAK 엔트리가 가지고 있는 일련 번호가 포함되어 전송되었다면, 상기 RLP

제어기 131은 취소 타이머의 값을 하나 줄인다. 만일 상기 재전송 타이머 값이 0이 되면, 상기 RLP제어기 131은 다음 동작을 수행한다. 만일 RLP제어기 131이 이미 NAK 엔트리가 가지고 있는 일련 번호에 해당하는 데이터 바이트를 수신하였다면, 상기 RLP제어기 131은 상기 NAK 엔트리를 삭제하고 상기 RLP제어기 131은 다음번 수신을 기다리는 데이터 프레임의 일련 번호로 L\_V(N) 레지스터 135를 설정한다. 이와 달리, RLP제어기 131이 데이터 바이트를 수신하지 못하였다면 상기 RLP제어기 131은 상기 NAK 엔트리의 취소 타이머를 적절한 값으로 설정한다. 상기 RLP제어기 131은 다음에 전송할 세 개의 NAK 프레임에 상기 NAK 엔트리가 가지고 있는 블록 일련 번호와 데이터 일련 번호를 포함시켜서 전송한다.

<333>      상기 RLP제어기 131은 새로 추가하여야 하는 NAK 엔트리들에 대해서 NAK 엔트리의 재전송 타이머를 적절한 값으로 설정하고, 다음에 전송할 두 개의 NAK 프레임에 상기 NAK 엔트리가 가지고 있는 일련 번호를 포함시켜서 전송한다.

#### 【발명의 효과】

<334>      상술한 바와 같이 본 발명에서는 데이터를 효과적으로 전송할 수 있도록 하기 위해 2차원 주소 방식의 일련 번호 방법을 제안하여 라디오 링크 프로토콜이 전송 효율을 높인다는 이점이 있다. 또한 본 발명에서는 상기 라디오 링크 프로토콜이 가변 길이의 프레임을 생성할 수 있도록 지원하는 다중화/역다중화기를 제안하는 이점이 있다.

**【특허청구범위】****【청구항 1】**

송신을 위한 데이터열을 저장하는 데이터 버퍼와,  
각각이 프레임 요구에 대응하는 크기 내의 데이터 블록들 또는 데이터 바이트들  
상기 데이터 버퍼로부터 읽어들이고 RLP프레임을 생성하여 송신하는 다수의 RLP처리기와,  
상기 다수의 RLP처리기중의 적어도 하나 이상의 RLP처리기로부터의 RLP프레임을 포  
합하는 가변 길이의 정보 비트를 생성하는 다중화 제어기와,  
상기 정보 비트를 물리채널을 통해 송신하는 물리계층 처리기를 포함함을 특징으로  
하는 이동 통신시스템의 데이터 송신기.

**【청구항 2】**

제1항에 있어서, 상기 프레임 요구는 상기 다중화 제어기에 의해 발생하는 것을 특  
징으로 하는 이동 통신시스템의 데이터 송신기.

**【청구항 3】**

제2항에 있어서, 상기 다중화 제어기는, 서비스의 우선순위에 따라 상기 다수의 서  
비스 제어기중의 해당하는 서비스 제어기로 상기 프레임 요구를 발생하는 것을 특징으로  
하는 이동 통신시스템의 데이터 송신기.

**【청구항 4】**

제1항에 있어서, 상기 다중화 제어기는, 서비스 인식자 및 길이 정보를 RLP프레임과 함께 포함하는 상기 정보 비트를 생성하는 것을 특징으로 하는 이동 통신시스템의 데이터 송신기.

**【청구항 5】**

제1항에 있어서, 상기 정보 비트는 기본 채널을 통해 송신되는 것을 특징으로 하는 이동 통신시스템의 데이터 송신기.

**【청구항 6】**

제1항에 있어서, 상기 정보 비트는 부가 채널을 통해 송신되는 것을 특징으로 하는 이동 통신시스템의 데이터 송신기.

**【청구항 7】**

제1항에 있어서, 상기 정보 비트는 적어도 하나 이상의 미리 설정된 논리 전송 단위의 정보 비트를 포함함을 특징으로 하는 이동 통신시스템의 데이터 송신기.

**【청구항 8】**

제1항에 있어서, 상기 정보 비트에는 적어도 하나 이상의 오류 검출을 위한 CRC비트가 포함됨을 특징으로 하는 이동 통신시스템의 데이터 송신기.

**【청구항 9】**

제1항에 있어서, 상기 다수의 RLP처리기로부터 송신되는 RLP프레임에 대한 블록 또는 바이트 단위의 일련 번호를 저장하는 레지스터를 더 포함함을 특징으로 하는 이동 통신시스템의 데이터 송신기.

**【청구항 10】**

제9항에 있어서, 상기 송신되는 RLP프레임의 각 블록 또는 바이트를 상기 일련 번호에 대응시켜 저장하는 재전송 버퍼를 더 포함함을 특징으로 하는 이동 통신시스템의 데이터 송신기.

**【청구항 11】**

제9항에 있어서, 상기 RLP처리기는 재전송 요구시 요구된 일련 번호에 대응하는 데이터 블록 또는 데이터 바이트를 상기 재전송 버퍼로부터 읽어들이어 재전송하는 것을 특징으로 하는 이동 통신시스템의 데이터 송신기.

**【청구항 12】**

할당된 물리채널을 통해 정보 비트를 수신하는 물리계층 처리기와,

상기 정보 비트로부터 적어도 하나 이상의 데이터 블록을 분리하는 역다중화 제어기와,

각각이 상기 역다중화 제어기에 의해 분리된 후 전달되는 RLP프레임을 블록단위 또는 바이트단위로 수신하는 다수의 RLP처리기와,

상기 다수의 RLP처리기에 수신된 RLP프레임 데이터를 저장하는 데이터 버퍼를 포함함을 특징으로 하는 이동 통신시스템의 데이터 수신기.

**【청구항 13】**

제12항에 있어서, 상기 역다중화 제어기는, 상기 분리된 데이터 블록과 함께 그 블록의 길이 정보를 해당하는 RLP처리기로 전달하는 것을 특징으로 하는 이동 통신시스템의 데이터 수신기.

**【청구항 14】**

제12항에 있어서, 상기 역다중화 제어기는, 기본 채널로 전송된 상기 정보 비트에 포함된 전송율과, 헤더정보를 이용하여 적어도 하나 이상의 데이터 블록 또는 데이터 바이트를 분리하는 것을 특징으로 하는 이동 통신시스템의 데이터 수신기.

**【청구항 15】**

제14항에 있어서, 상기 역다중화 제어기는, 상기 정보 비트가 깨진 경우, 상기 다수의 서비스 제어기에 깨진 데이터 블록 또는 데이터 바이트가 수신되었음을 알리는 것을 특징으로 하는 이동 통신시스템의 데이터 수신기.

**【청구항 16】**

제12항에 있어서, 상기 역다중화 제어기는, 논리 전송 단위가 사용된 상기 정보 비트가 부가 채널로 전송된 경우, 상기 정보 비트의 각 논리 전송 단위로 포함된 서비스 인식자를 이용하여 데이터 블록 또는 데이터 바이트를 분리한 후 해당하는 RLP처리기로 전달하는 것을 특징으로 하는 이동 통신시스템의 데이터 수신기.

**【청구항 17】**

제12항에 있어서, 상기 역다중화 제어기는, 논리 전송 단위가 사용되지 않은 상기 정보 비트가 부가 채널로 전송된 경우, 상기 정보 비트에 포함된 서비스 인식자를 이용하여 데이터 블록 또는 데이터 바이트를 분리한 후 해당하는 RLP처리기로 전달하는 것을 특징으로 하는 이동 통신시스템의 데이터 수신기.

**【청구항 18】**

적어도 하나 이상의 서비스들의 전송 순서와, 데이터 블록의 크기를 결정하는 과정과,

상기 서비스들의 데이터 블록을 요구하는 과정과,

상기 요구에 따른 적어도 하나 이상의 데이터 블록 또는 데이터 바이트를 모아 가변 길이의 정보 비트를 생성하는 과정과,

상기 생성된 정보 비트를 물리채널을 통해 송신하는 과정을 포함함을 특징으로 하는 이동 통신시스템의 데이터 송신방법.



**【청구항 19】**

제18항에 있어서, 상기 서비스들의 우선순위에 따른 상기 요구가 발생하는 것을 특징으로 하는 이동 통신시스템의 데이터 송신방법.

**【청구항 20】**

제18항에 있어서, 상기 정보 비트는 서비스 인식자 및 길이 정보가 상기 데이터 블록 또는 데이터 바이트와 함께 포함된 것임을 특징으로 하는 이동 통신시스템의 데이터 송신방법.

**【청구항 21】**

제18항에 있어서, 상기 정보 비트는 기본 채널을 통해 송신되는 것을 특징으로 하는 이동 통신시스템의 데이터 송신방법.

**【청구항 22】**

제18항에 있어서, 상기 정보 비트는 부가 채널을 통해 송신되는 것을 특징으로 하는 이동 통신시스템의 데이터 송신방법.

**【청구항 23】**

제18항에 있어서, 상기 정보 비트는 적어도 하나 이상의 미리 설정된 논리 설정 단위의 정보 비트를 포함함을 특징으로 하는 이동 통신시스템의 데이터 송신방법.

【청구항 23】

【청구항 24】

제18항에 있어서, 상기 정보 비트에는 적어도 하나 이상의 오류 검출을 위한 CRC비트가 포함됨을 특징으로 하는 이동 통신시스템의 데이터 송신방법.

【청구항 25】

할당된 물리채널을 통해 정보 비트를 수신하는 과정과,

상기 정보 비트로부터 적어도 하나 이상의 데이터 블록 또는 데이터 바이트를 분리하는 과정과,

상기 분리되는 각 데이터 블록 또는 데이터 바이트를 다수의 서비스중 어느 한 서비스를 위해 전달하는 과정을 포함함을 특징으로 하는 이동 통신시스템의 데이터 수신 방법.

【청구항 26】

제25항에 있어서, 상기 분리되는 각 데이터 블록 또는 데이터 바이트와 함께 그 길이 정보를 함께 전달하는 것을 특징으로 하는 이동 통신시스템의 데이터 수신 방법.

**【청구항 27】**

제25항에 있어서, 상기 정보 비트는 기본 채널을 통해 전송된 정보 비트이며, 이 정보 비트에 포함된 전송율과, 헤더정보를 이용하여 적어도 하나 이상의 데이터 블록 또는 데이터 바이트를 분리하는 것을 특징으로 하는 이동 통신시스템의 데이터 수신 방법.

**【청구항 28】**

제27항에 있어서, 상기 정보 비트가 깨진 경우, 깨진 데이터 블록 또는 데이터 바이트가 수신되었음을 상기 다수의 서비스를 위해 전달하는 것을 특징으로 하는 이동 통신시스템의 데이터 수신 방법.

**【청구항 29】**

제27항에 있어서, 상기 정보 비트가 깨지지 않고, 하나의 데이터 블록 또는 데이터 바이트만이 수신된 경우, 널트래픽으로 간주하고 상기 정보 비트를 버리는 것을 특징으로 하는 이동 통신시스템의 데이터 수신 방법.

**【청구항 30】**

제27항에 있어서, 상기 정보 비트가 깨지지 않은 경우, 빈 데이터 블록이 수신되었음을 상기 다수의 서비스중 수신된 데이터를 처리하지 않는 서비스를 위해 전달하는 것

을 특징으로 하는 이동 통신시스템의 데이터 수신 방법.

**【청구항 31】**

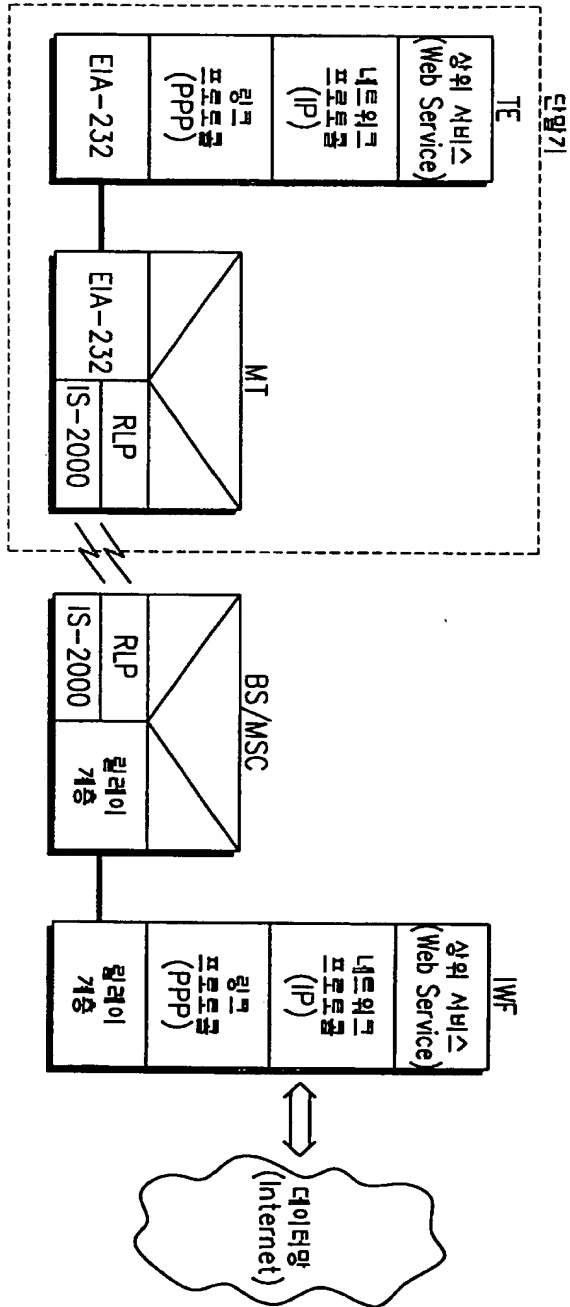
제29항에 있어서, 상기 정보 비트는 논리 전송 단위로 사용되어 부가 채널을 통해 전송된 정보 비트이며, 이 정보 비트의 각 논리 전송 단위로 포함된 서비스 인식자를 이용하여 데이터 블록 또는 데이터 바이트를 분리하는 것을 특징으로 하는 이동 통신시스템의 데이터 수신 방법.

**【청구항 32】**

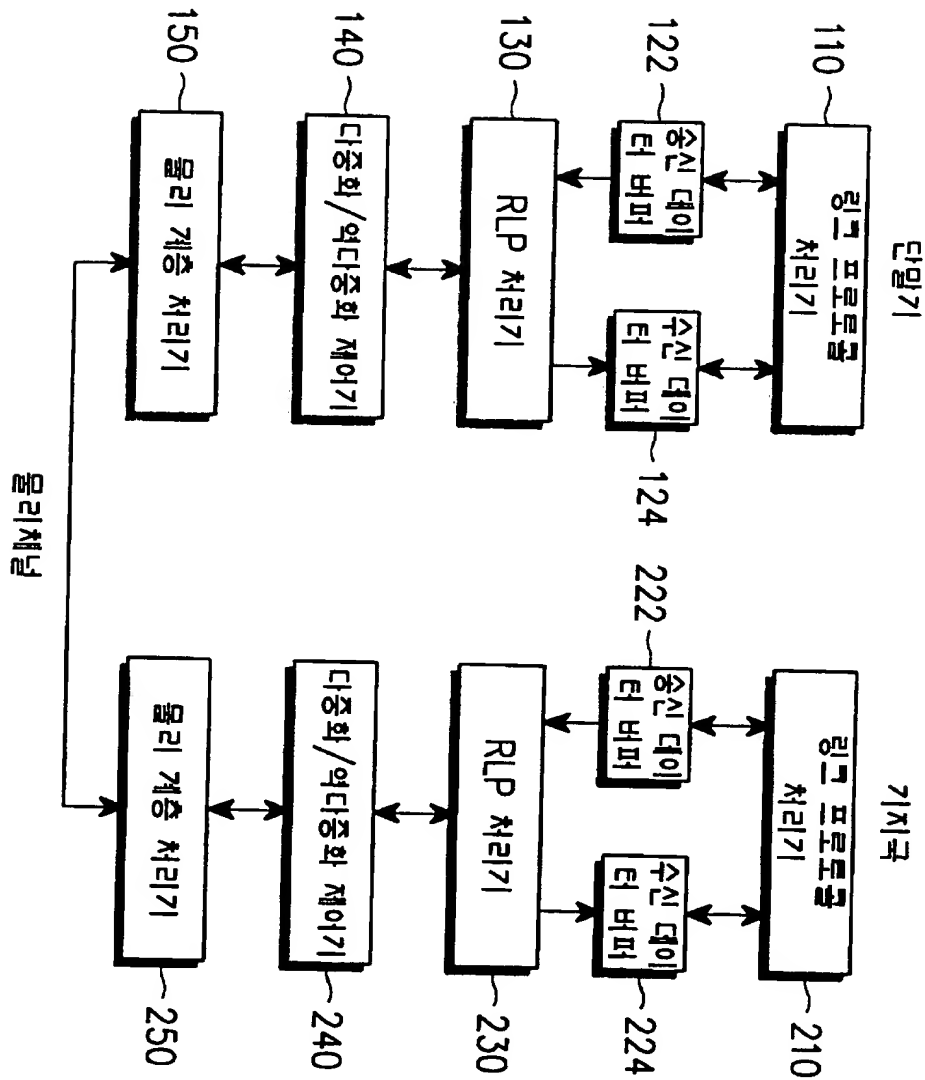
제25항에 있어서, 상기 정보 비트는 논리 전송 단위가 사용되지 않고 부가 채널을 통해 전송된 정보 비트이며, 이 정보 비트에 포함된 서비스 인식자를 이용하여 데이터 블록 또는 데이터 바이트를 분리하는 것을 특징으로 하는 이동 통신시스템의 데이터 수신 방법.

【도면】

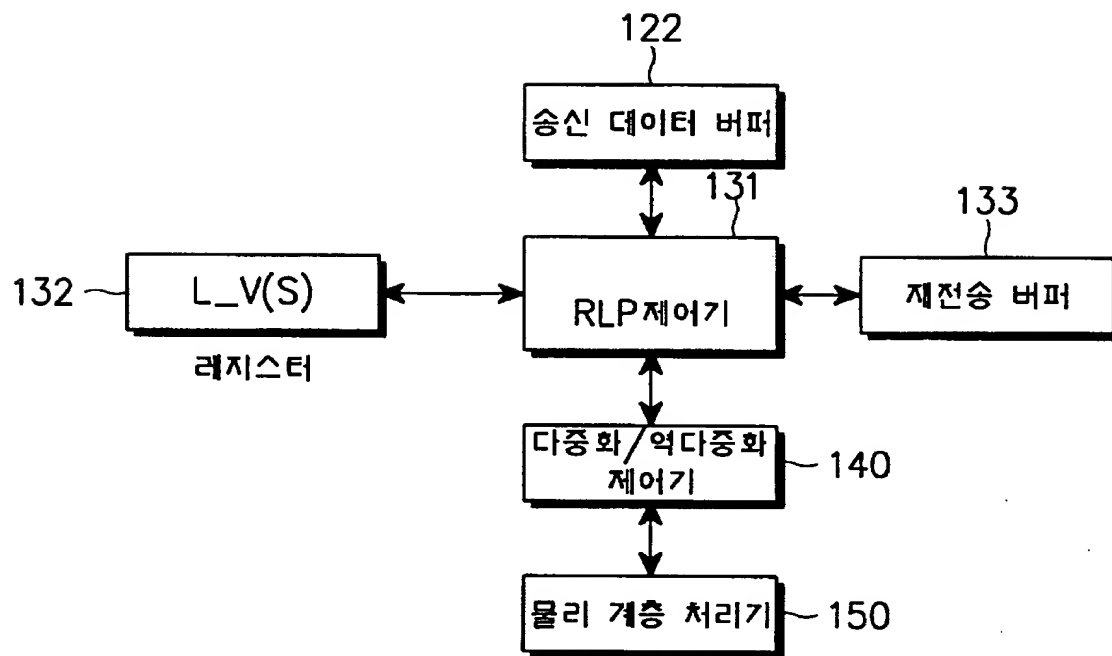
【도 1】



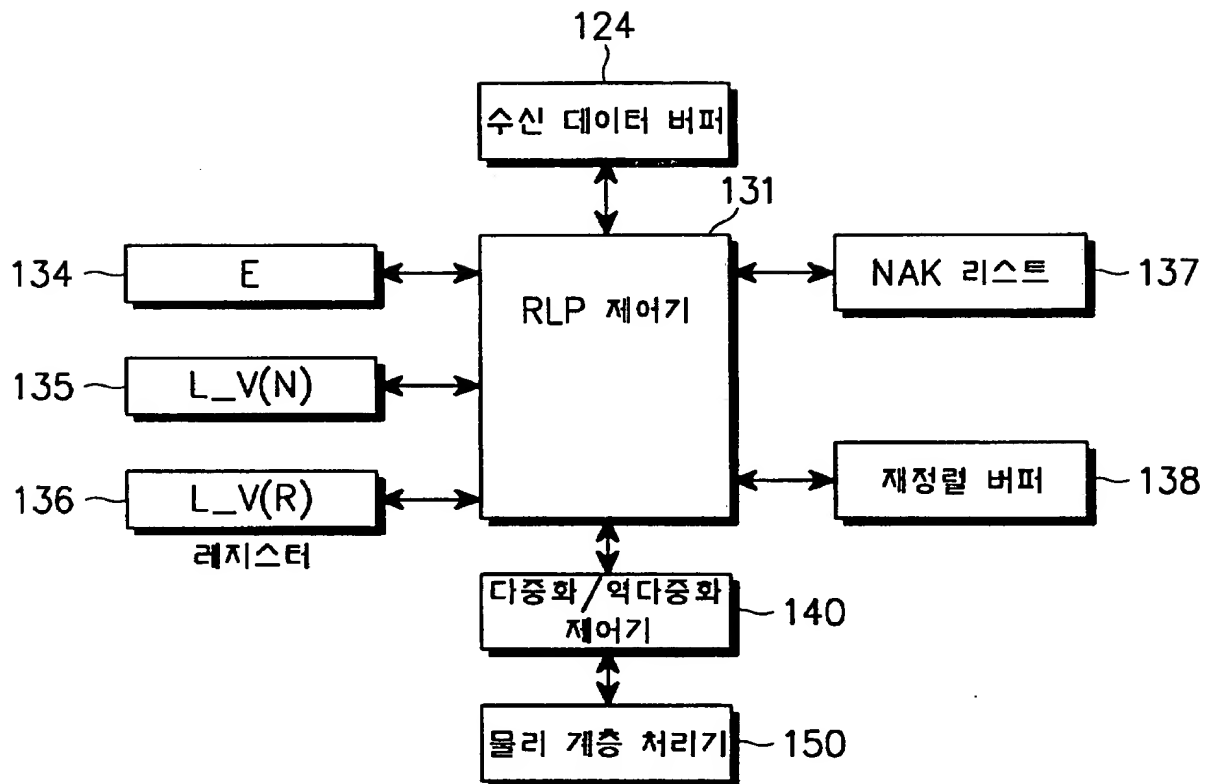
【도 2】



【도 3】

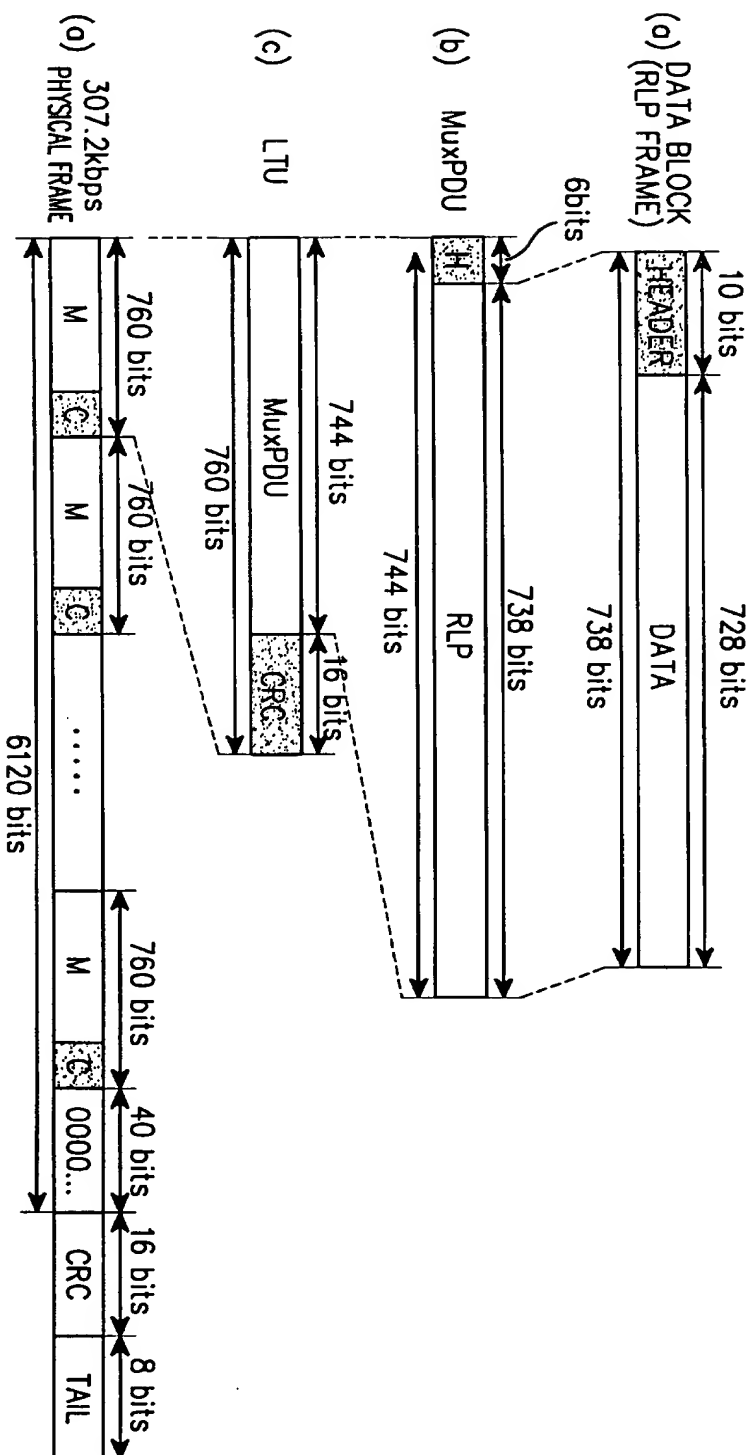


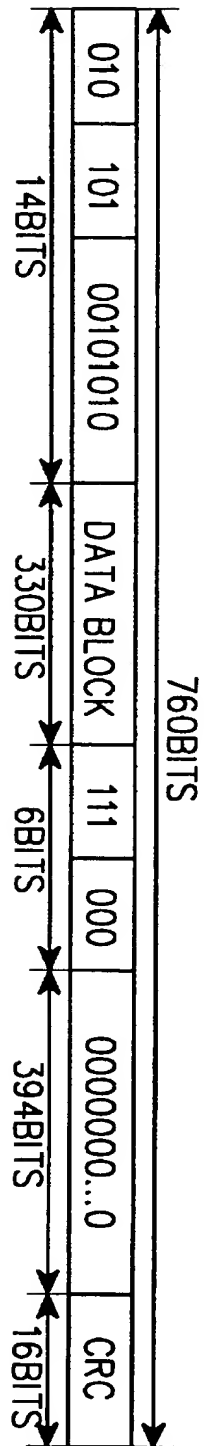
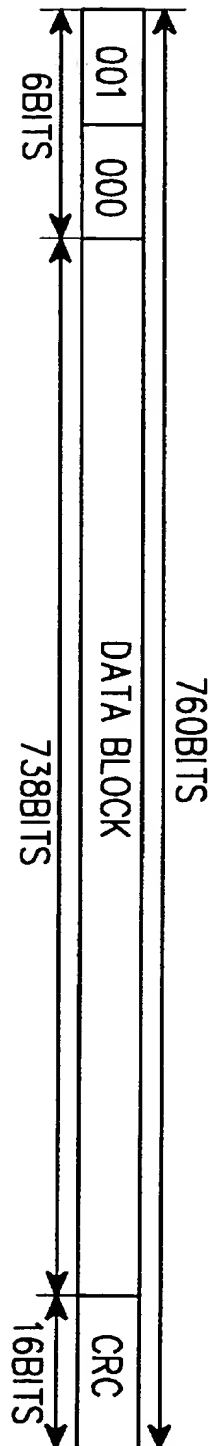
【도 4】



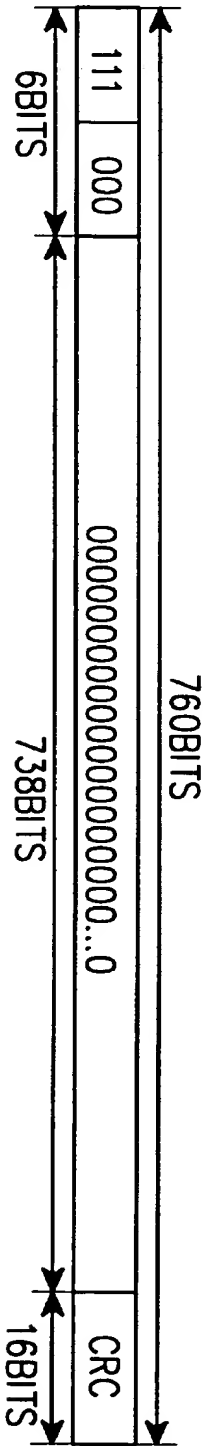


【 5 】

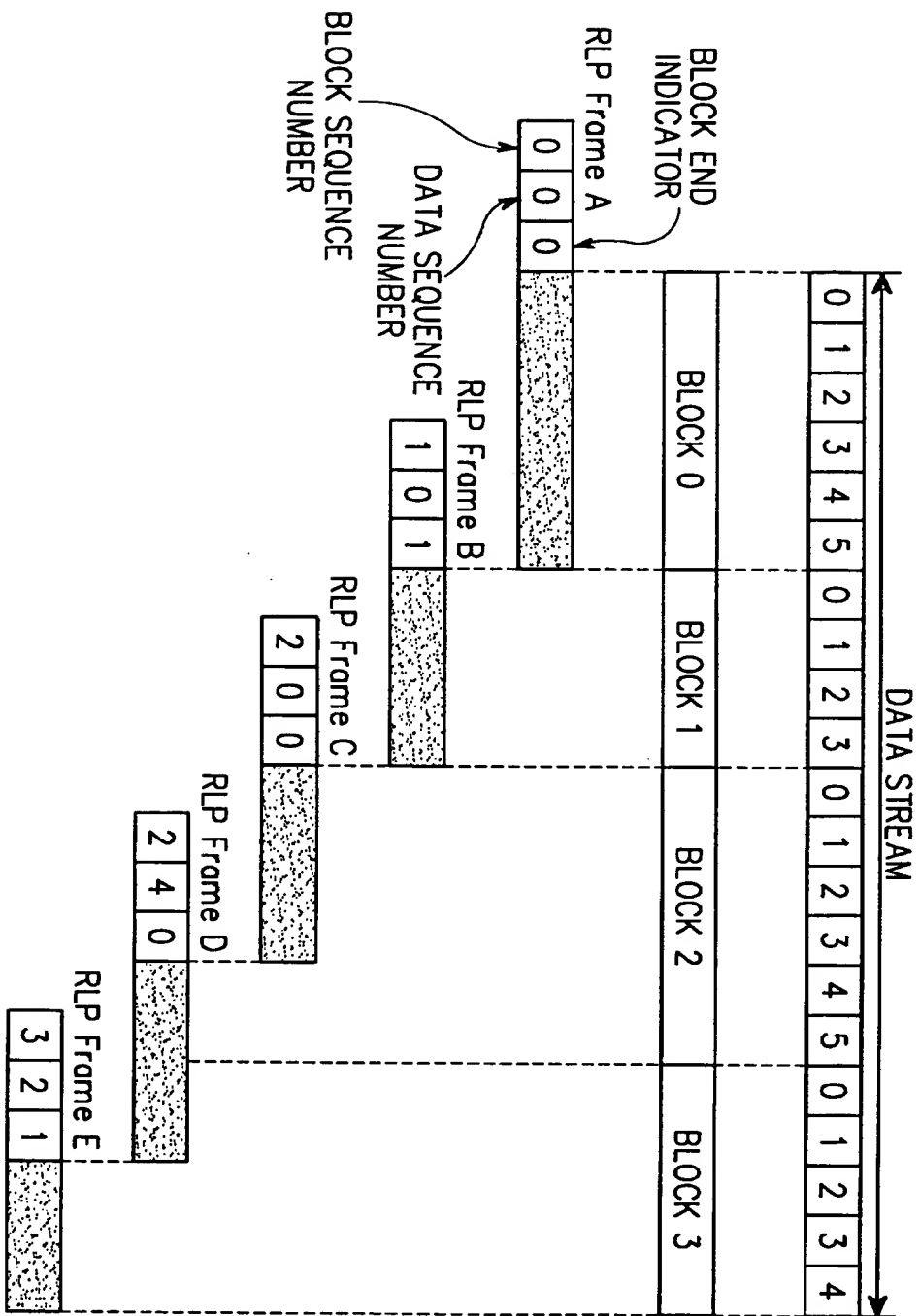




【图 6】

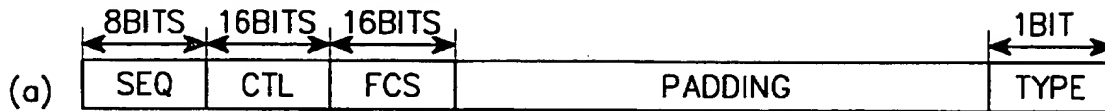


【도 7】



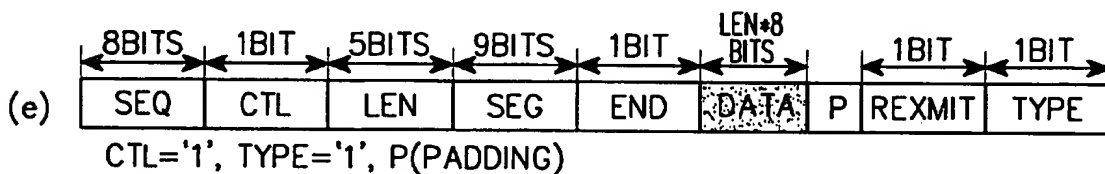
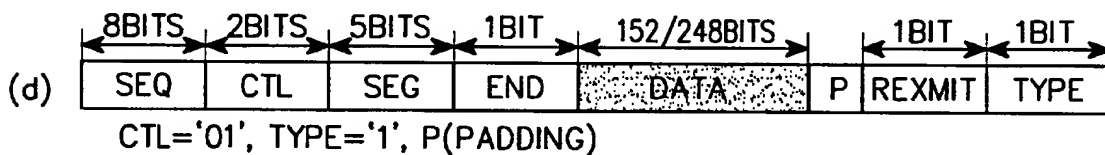
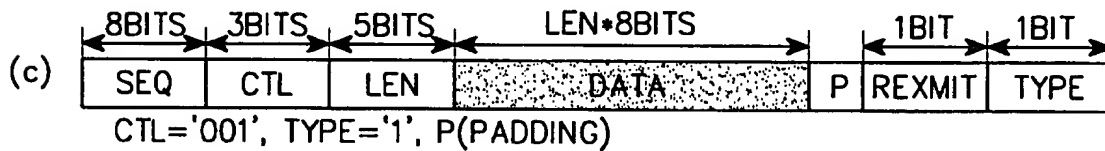
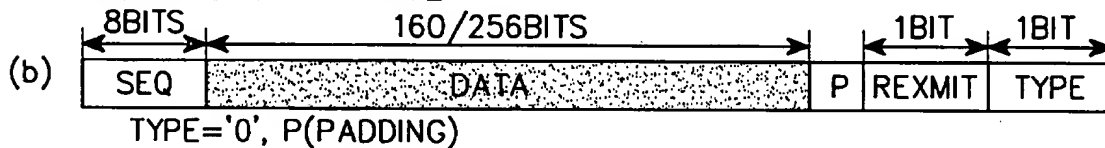
## 【도 8】

## SYNC, SYNC/ACK, ACK FRAME

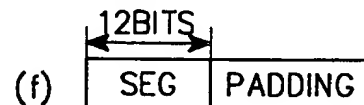


CTL='00011001' FOR SYNC, '00011010' FOR  
 SYNC/ACK, '00011011' FOR ACK, TYPE='1'

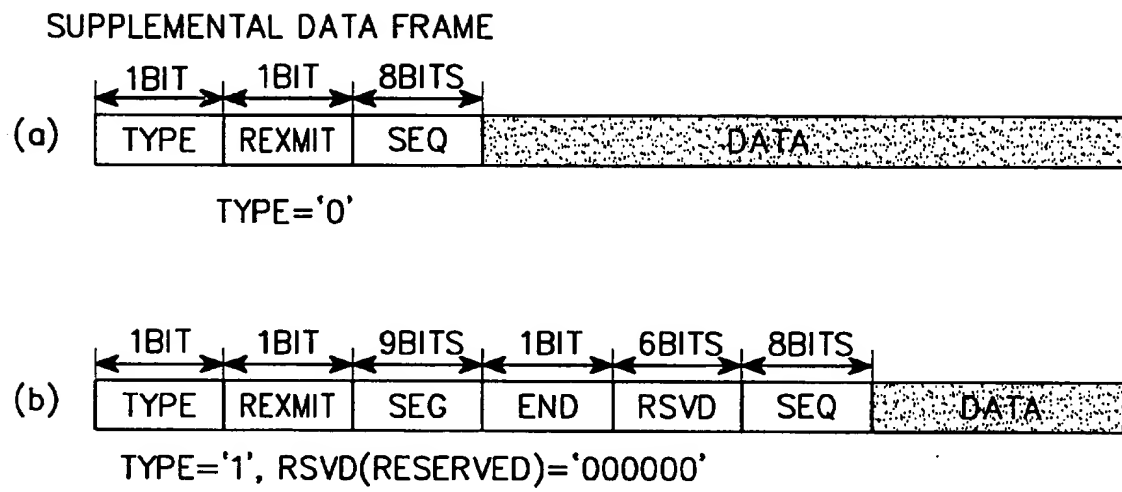
## FUNDICATED DATA FRAME



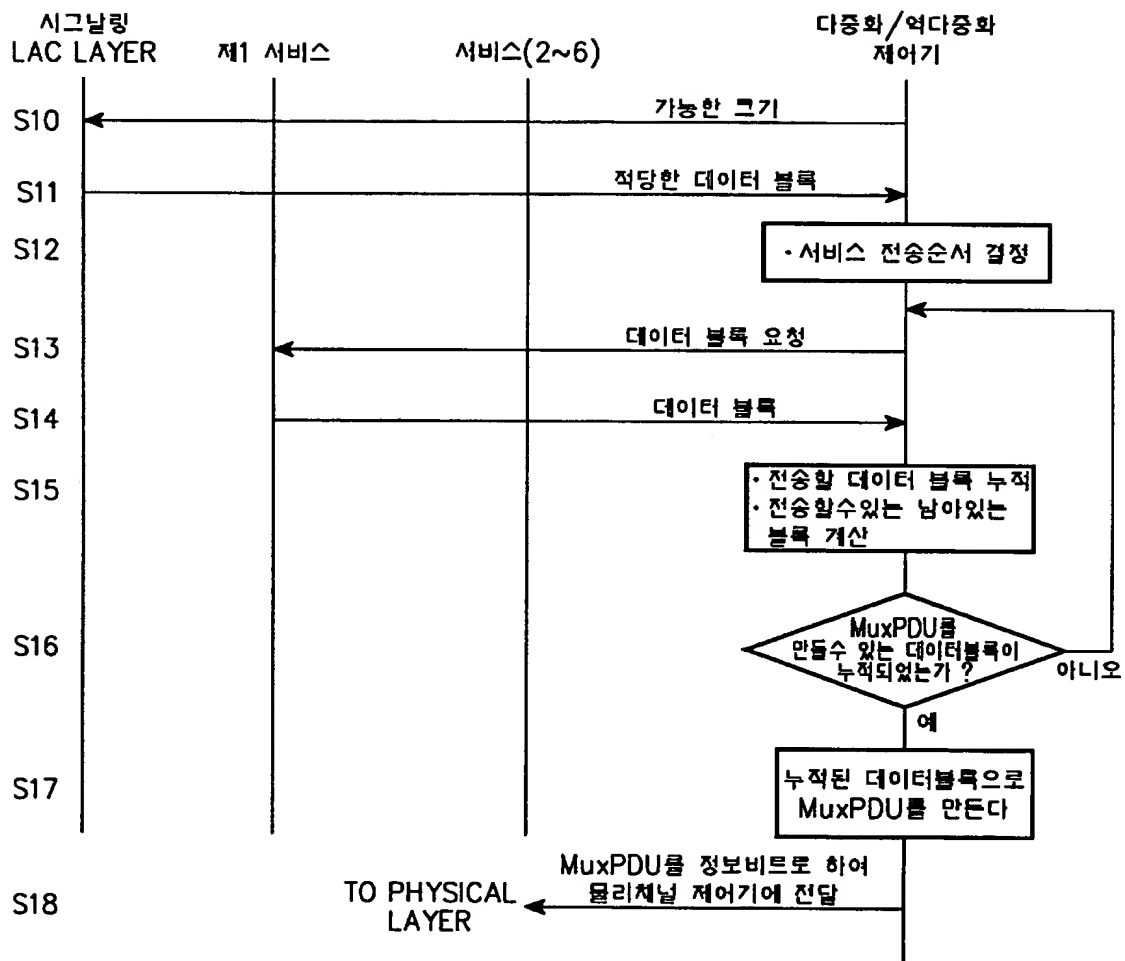
## IDLE FRAME(1/8 RATE ONLY)



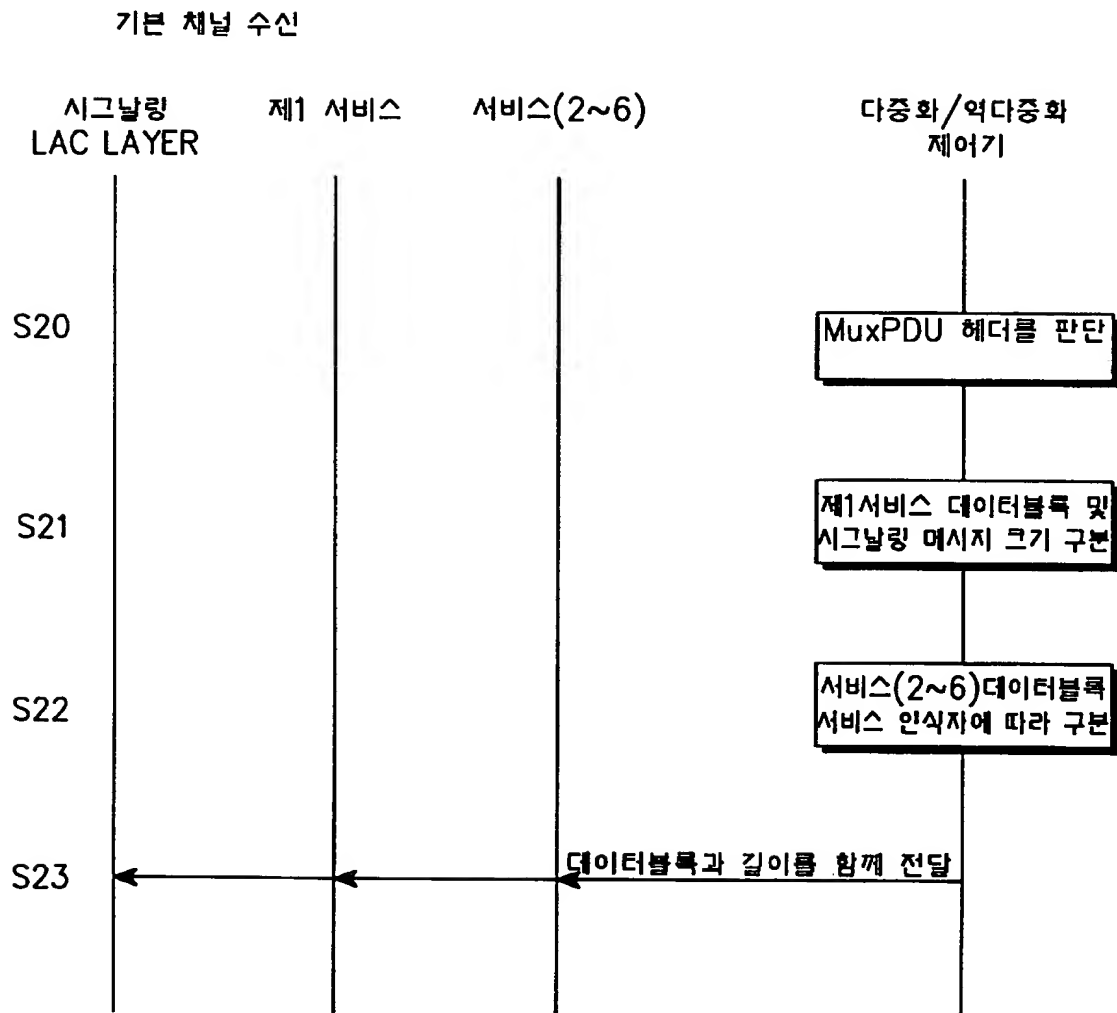
【도 9】



【도 10】

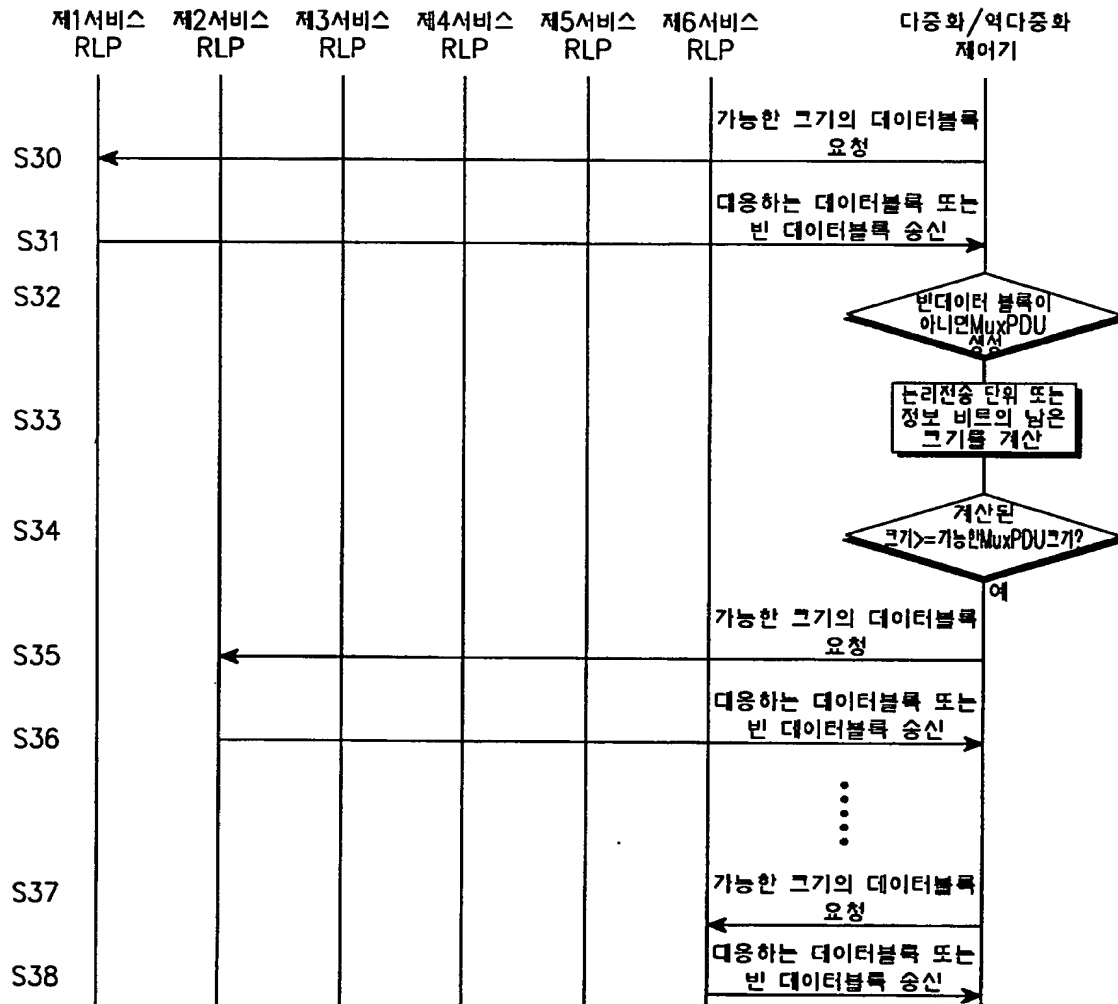


【도 11】



【도 12】

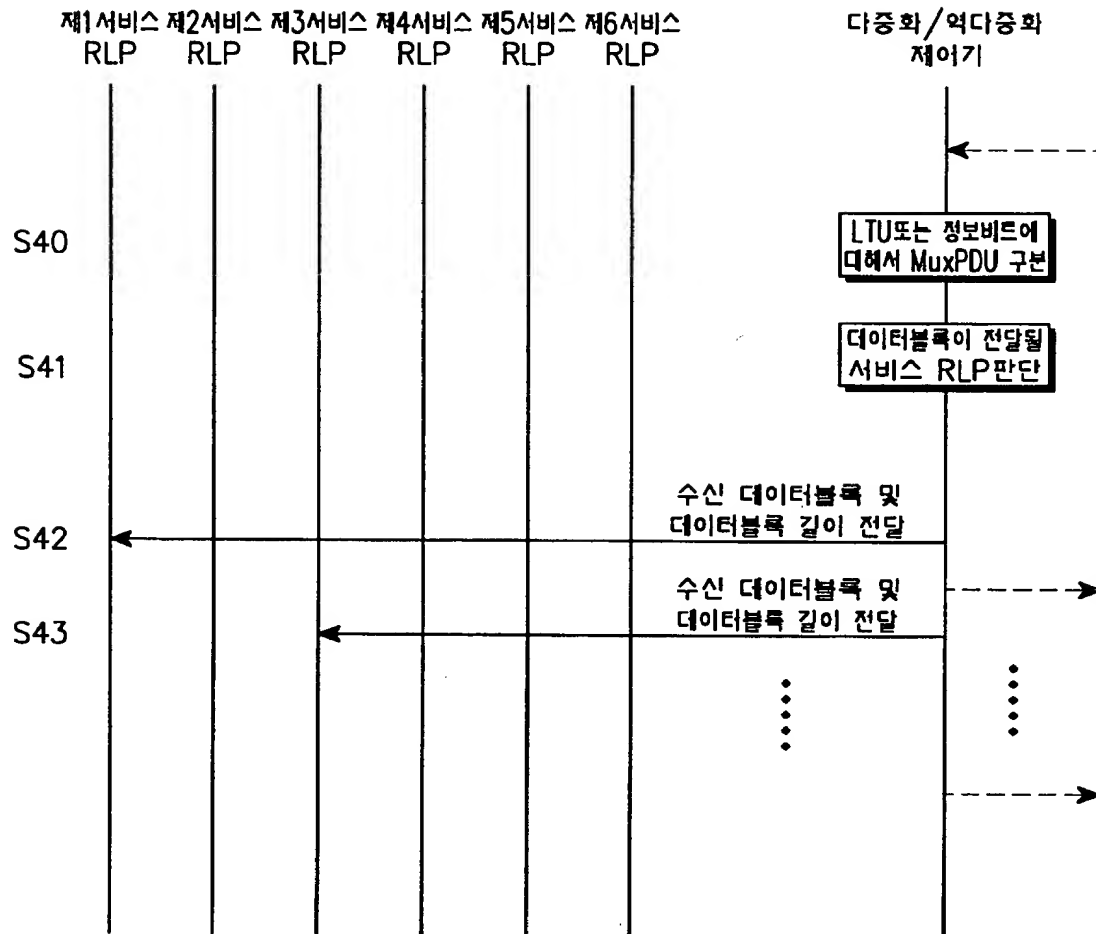
부가 채널 송신





【도 13】

부가 채널 수신



【서류명】	서지사항보정서
【수신처】	특허청장
【제출일자】	1999.06.28
【제출인】	
【명칭】	삼성전자 주식회사
【출원인코드】	119981042713
【사건과의 관계】	출원인
【대리인】	
【성명】	이건주
【대리인코드】	919980003398
【사건의 표시】	
【출원번호】	1019990020081
【출원일자】	1999.05.27
【발명의 명칭】	이동통신시스템에서라디오링크프로토콜에따른가변길이의데 이터송수신장치및방법
【제출원인】	
【발송번호】	151999001509278
【발송일자】	1999.06.17
【보정할 서류】	특허출원서
【보정할 사항】	
【보정대상 항목】	수수료
【보정방법】	납부
【보정내용】	
【수수료】	미납 수수료
【취지】	특허법시행규칙 제13조·실용신안법시행규칙 제12조의 규정 에 의하여 위와 같이 제출합니다.
【수수료】	
【보정료】	11000
【기타 수수료】	129000
【합계】	140000